Departamento de Informática

PROYECTO FIN DE CARRERA

Modelo para la actualización eficiente de contenidos en un Crawler de ficheros RSS

Autor: José Vicente Sevillano Martín

Tutor: Isabel Segura Bedmar

Leganés, octubre de 2011

Título: **¡Error! No se encuentra el origen de la referencia.**
Autor: José Vicente Sevillano Martín
Director: Isabel Segura Bedmar

EL TRIBUNAL

Presidente: Dolores Cuadra

Vocal: Harith Al-Jumaily

Secretario: María González García

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 17 de octubre de 2011 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

Agradezco a Isabel Segura, tutora del proyecto, y a César de Pablo, principal valedor del mismo, su apoyo y ayuda en todo momento. Ellos han sido los artífices de que se haya conseguido cumplir un propósito que se ha alargado en el tiempo más de lo deseado pero por el que al final ha merecido la pena el sacrificio.

También agradecer la infinita paciencia que ha tenido conmigo mi novia, Carmen, a quien he tenido indiscriminadamente castigada con la indeterminación de la meta en la que tantas ilusiones tenía puestas en mí, y que espero no haber decepcionado.

Resumen

El proyecto que se expone a continuación tiene por objetivo la mejora de la eficiencia del aplicativo conocido como Crawler, el cual se construyó con la finalidad de recuperar actualizaciones de ficheros de sindicación de sitios Web. Esta actualización se realiza visitando distintas páginas en Internet de manera periódica y en caso de existir datos nuevos recuperarlos. La manera que tiene de recuperar dichas actualizaciones, visitando en cada lanzamiento del Crawler cada página Web indicada en un fichero, hace que la eficiencia del proceso no sea la adecuada debido a la alta probabilidad de que un sitio Web no haya sido actualizado desde la última visita que se realizó. Con este proyecto se pretende programar en cierta manera los lanzamientos del Crawler para que cuando se realicen, su tasa de aciertos sea mucho mayor de lo que es en el sistema actual.

El nuevo sistema priorizará páginas Web teniendo en cuenta el histórico de actualizaciones para valorar si en un lanzamiento del Crawler sería conveniente visitar cierta página porque se espera que esté actualizada.

Debido a la mejora en la precisión y que, por tanto, acarrea menos trabajo del procesador, se podrá hacer crawling de muchas más fuentes.

Palabras clave: Crawler, robot, araña, rss, feed, sindicador, priorizador, poisson

Abstract

The main objetive of the project expounded here is the improvement in the efficiency of application known as Crawler, wich was built with the aim of retrieving updates from websites syndicating files. This update is executed checking different web pages periodically and retreving new data when they are found. The way it recovers those updates is visiting every web indexed in a file at every Crawler's launching, which is not the most adequate process because it is high probable that a website has not been updated since the last visit was performed.

This project aims to program Crawler's launchings in order to improve its success rate. The new system gives priority to certain websites taking into account the historcal of updates to decide whether the Crawler must visit a specific site which is expected to be updated.

A higher precision entails less work for the processor so it is possible to crawl much more sources.

Keywords: Crawler, robot, spider, rss, feed, syndication, prioritizer, poisson

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y objetivos

1.1 Introducción

El proyecto que se desarrolla en estas líneas se ha diseñado como complemento a un sistema realizado con anterioridad, que es utilizado para recuperar ficheros de sindicación de páginas Web y procesarlos de manera automática y al que se conoce como Crawler. Este proyecto servirá para solventar ciertos problemas que se producen en el Crawler y que, a pesar de no ser un impedimento para su funcionamiento, sí que incurren en un grado de explotación del procesador innecesario para la tarea encomendada.

El sistema inicial es requerido para localizar y descargar contenidos de determinados lugares, concretamente ficheros de sindicación de contenidos en la Web, para su posterior procesamiento. Dicho sistema se ha denominado Crawler y está enmarcado dentro de un sistema mayor, que tiene por finalidad reunir información política de forma automática y constante.

Se pretende hacer un seguimiento de lo que ocurre en cada momento a nivel político en nuestro país. Inicialmente, la idea es que el sistema sea capaz de gestionar de forma automática el contenido, enlazar contenido similar, asociar fuentes y políticos con noticias y otras funcionalidades similares. El objetivo es que este sistema sea capaz de trabajar de forma autónoma, descargado constantemente el contenido de una lista de Webs proporcionada y almacenando ese contenido en una serie de ficheros de salida.

El Crawler funciona de manera que realiza una serie de accesos a las distintas páginas Web definidas en un fichero. Se realiza un acceso cada vez que es lanzado. Se lanza una vez cada quince minutos. El problema detectado en este procedimiento es que no se recuperan entradas cada vez que se realiza un acceso a la Web por cada lanzamiento del Crawler. La mayoría de las veces que se accede a una Web no se obtiene ninguna entrada nueva. Es por esto que se realiza un trabajo innecesario que, sin embargo, está repercutiendo en el funcionamiento del procesador y el trabajo del servidor en el que esté alojada la aplicación en general.

El objetivo de este proyecto es el de diseñar e implementar un modelo que de solución a este problema, intentando que el Crawler sea lanzado el mínimo de veces posible asegurando la recuperación de las entradas.

1.2 Objetivos

El objetivo principal de este proyecto es el de diseñar e implementar un modelo que consiga que el Crawler sea lanzado el mínimo de veces posible, asegurando además la recuperación de las entradas con contenido relevante, es decir, que no se hayan recuperado con anterioridad.

Debido a que no siempre que el Crawler es lanzado se estarán actualizados todos los sitios Web, se realizará una estimación de qué sitios es posible que estén actualizados. Los sitios que tengan una probabilidad alta serán visitados en la siguiente pasada y los que tengan una probabilidad baja serán visitados más adelante o directamente no serán visitados.

Además del objetivo principal, se va hacer hincapié en la consecución de algunos objetivos secundarios, que se ha creído conveniente desarrollar para la implementación del modelo priorizado:

Uso de conocimientos adquiridos

Para la realización del diseño que se propone como solución en el presente proyecto, se ha querido utilizar los conocimientos logrados en los propios estudios. Con ello se pretende demostrar que estos conocimientos no sólo son válidos en el contexto del estudio universitario, sino que es posible aplicar lo aprendido en los futuros proyectos que se puedan abordar.

El diseño del modelo priorizado se ha basado en algunos fragmentos de estadística descriptiva, concretamente en los modelos de probabilidad univariantes, para estimar la probabilidad de que un suceso ocurra.

Gestión de errores

Se debe construir un sistema robusto. Esto significa, que el sistema no debe detenerse por ciertos errores producidos en tiempo de ejecución. Los problemas ocasionados por un mal funcionamiento del módulo de estadísticas no debe impedir el funcionamiento del Crawler. En este sentido, en caso de no poder producir las entradas a partir de las estadísticas recuperadas, se deberán realizar las visitas que se hubieran hecho con el modelo antiguo del Crawler.

Casos fuera del modelo

Se debe tener en cuenta que habrá casos que no abarque el modelo. Por tanto habrá sitios Web que no serán incluídos nunca en los OPML de entrada para su visita por el Crawler. Sin embargo, estos sitios Web deben ser visitados también. Por ello se debe especificar una solución para estos casos que no abarque el modelo.

1.3 Fases del desarrollo

Análisis. Para realizar este proyecto ha sido necesaria la elección de un modelo adecuado para conseguir el objetivo principal. Dicho modelo ha sido escogido de entre varias alternativas analizadas, considerando las ventajas y los inconvenientes que pudieran tener para abordar el proyecto con garantías de fiabilidad. El modelo escogido está basado en una combinación lineal de distribuciones de Poisson, que interpretan conjuntamente la distribución de las entradas recuperadas por el Crawler. Al modelo final se le ha denominado modelo priorizado, debido a que dará prioridad a unos sitios Web, mientras que marginará a otros.

Simulación. El modelo elegido ha sido probado mediante un simulador implementado también como parte del estudio. El simulador ha sido realizado en un libro Excel y abarca una cantidad de datos suficientes para comprobar el funcionamiento del modelo. Se han modificado distintos parámetros para afinar la fórmula que lleva asociada la priorización, de manera que las distintas simulaciones mostraran unos resultados más normalizados.

1.4 Medios empleados

Para la realización del proyecto se han utilizado varios dispositivos para ejecutar las distintas tareas.

El diseño y la implementación del modelo se han realizado en un portátil con procesador Intel Core 2 Duo, 2,2 GHz y una memoria RAM de 2 GB. El entorno de desarrollo ha sido el IDE Eclipse en su versión 3.5.2, con la versión 1.5 del jdk de java como compilador. El gestor de bases de datos utilizado es MySQL en la versión 5.0. La simulación del modelo se ha implementado en un libro Excel del paquete Microsoft Office XP.

Para la recuperación de datos se ha utilizado un ordenador de sobremesa con procesador Pentium IV, 800 MHz y una memoria RAM de 512 GB. Con la versión 1.5 del jre de java instalado, además de la versión 5.0 de MySQL.

1.5 Estructura de la memoria

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo:

- **Introducción y objetivos.** En este capítulo se explica la motivación del presente proyecto, así como los objetivos que se han querido lograr para la realización del mismo.

- **Estado del arte.** Una pequeña muestra de qué aplicación se está tratando y los servicios o aplicativos que se pueden encontrar en la actualidad.

- **Diseño del modelo priorizado.** Se expone la realización del diseño del modelo. Las opciones barajadas y el por qué de las decisiones tomadas.

- **Simulación del modelo.** Un resumen de la simulación del comportamiento del modelo, realizada con un libro Excel, y los resultados obtenidos.

- **Implementación.** La explicación de cómo se ha implementado el modelo a partir del diseño y pruebas realizadas con los datos de la simulación.

- **Experimentación.** Se muestran las pruebas realizadas con datos reales aplicados al modelo implementado.

- **Conclusiones.** Una breve explicación del punto en el que se partía, a dónde se ha llegado y qué pasos se podrían dar a continuación.

- **Costes y presupuesto.** Se ha calculado un presupuesto acorde a los costes asociados al diseño e implementación del modelo. Así como su implantación.

- **Glosario.** Se expone una definición de términos que aparecen en el presente documento.

- **Referencias.** Algunas referencias consultadas a la hora de elaborar el proyecto.

Capítulo 2

Estado del arte

2.1 Web Crawler

Un Web Crawler es un sistema capaz de recorrer, de forma planificada, un conjunto de recursos. Estos recursos pueden ser locales o remotos, por ejemplo URL´s. Generalmente, solo descarga los contenidos si se consideran relevantes para su objetivo. Además, por definición, un web crawler es capaz de descubrir recursos relacionados y relevantes para la aplicación.

El uso más común de los Web Crawler es el de motor de búsqueda. No obstante, el web Crawler más famoso y utilizado del mundo es el del buscador Google, conocido también como Google Spider o Google Bot. Este sistema es capaz de ir descargando y almacenando la información de cientos de miles de sitios web en unas horas.

En la actualidad, la utilidad de este tipo de sistemas está muy extendida. Más allá de los típicos buscadores de internet, es posible encontrar sistemas de bussines

intelligence, sistemas capaces de recopilar y en la mayoría de los casos analizar información de la competencia o de un determinado mercado.

Este no es más que un ejemplo de uso de un Crawler. Aunque también hay usos más fraudulentos, como la utilización de esta clase de sistemas para la obtención de direcciones de correo electrónico, datos personales o datos bancarios.

Un amplio desarrollo sobre Web Crawling puede encontrarse en el capítulo 8 del libro Web Data Mining que se encuentra enlazado al final del documento.

2.2 Lector RSS

Un lector RSS es un programa que permite a una persona darse de alta en las RSS de sus páginas Web o blogs favoritos para recibir los artículos y contenidos que son de su interés.

Los lectores RSS reúnen, en un solo lugar, todos los titulares de las páginas Web a las que se ha suscrito el usuario. En función del Lector RSS elegido, el programa o lector online ofrecerá al usuario la opción de organizar la información por carpetas o por categoría según sus preferencias o interés. Si un usuario está suscrito a páginas Web sobre hobbies y sobre aspectos relacionados con su profesión, el usuario podría organizar la información por pestañas, es decir, una pestaña para las páginas sobre sus hobbies (por ejemplo, coches) y otra pestaña para las páginas de interés profesional (por ejemplo, publicidad).

2.2.1 Lectores RSS en la Web

Existen distintos tipos de lectores RSS según su arquitectura.

Instalados directamente en el ordenador: Son programas que se instalan en cada ordenador. Cuando está abierto, el programa accede cada cierto tiempo a las páginas web suscritas para traer las actualizaciones directamente al ordenador. Un programa conocido es Feedreader.

Online: Los lectores Rss online cumplen la misma función que los programas que se instalan en el ordenador, aunque en este se hace todo a través de una página Web. Para ello, el usuario se tiene que dar de alta en la página Web que ofrece ese servicio y dar de alta un perfil. A partir de ese momento, se puede acceder en cualquier momento al lector Web introduciendo el nombre de usuario y contraseña. Un programa conocido es Google Reader.

Navegador Web o programa de correo electrónico: También se puede recibir las actualizaciones de las páginas Web a través del navegador Web o del programa de correo electrónico. Algunos de los navegadores y clientes de correo más conocidos que permiten hacer esto son: Internet Explorer, Mozilla Firefox, Outlook Express o Mozilla Thunderbird.

2.3 Readers Open Source

Existen distintos lectores RSS que podrían haber ahorrado el esfuerzo de crear un modelo de priorización, como pueden ser GoogleReader, RSSOwl o Nutch. Estos lectores de RSS tienen distintas ventajas.

Por un lado, GoogleReader, es un lector de RSS online, con lo que realiza la tarea de visitar la Web por él mismo. Así, bastaría con incluirle una lista de canales (el fichero OPML de entrada) para que buscara siempre si se han producido actualizaciones. Dichas actualizaciones estarían siempre disponibles en GoogleReader, por lo que se podría acceder por ejemplo cada hora a esas actualizaciones y descargarlas, en caso de haberlas. Una tarea, a priori, bastante sencilla y que realizaría prácticamente el trabajo completo. El problema de GoogleReader es que tiene un límite de 2000 canales. Puede parecer un límite muy elevado, pero no se sabe a priori como crecerá el número de canales en la aplicación actual. Además, con herramientas de terceros, siempre se tiene la posibilidad de que en un momento dado deje de funcionar o se modifiquen los términos de licencia, con lo que desde ese momento el Crawler no podría recuperar entradas.

Por otra parte, RSSOwl es un lector RSS instalado en el ordenador. En su caso se puede integrar en un proyecto Java. Pero al final realiza la misma acción que está realizando el Crawler que ya se tiene desarrollado. Es decir, que al final se tendría que realizar también el módulo de estadísticas para que no realizara siempre las visitas a todos los canales.

Nutch es un motór de búsqueda Web genérico, que principalmente está destinado a recuperación de información de páginas HTML, aunque también puede configurarse para recuperar información de otro tipo de ficheros, entre ellos de sindicación de contenido.

Está escrito en Java y basado en Lucene, que es un API de código abierto para recuperación de información. Podría ser configurado para lanzarse en un equipo, aunque ellos mismos recomiendan lanzarlo en un clúster Hadoop, que es un framework propio de apache.

El gran problema de Nutch es que habría que configurar todo su sistema y después adaptar el Crawler para crear un feedback con este motor.

Teniendo en cuenta estas características se ha decidido implementar sobre el Crawler que ya se disponía el módulo de estadísticas que implementa el modelo priorizado diseñado en el presente documento.

Capítulo 3

Diseño del modelo priorizado

3.1 Introducción

En este apartado se explicarán los pasos a seguir para la realización de un diseño que solucione el problema planteado de la manera más eficiente. Se podrán conocer los modelos analizados para intentar recuperar la mayor información posible sin necesidad de visitar continuamente todas las páginas de sindicación que se requiera. Se expondrán las razones del rechazo de distintos modelos o de la mejora del modelo que concordaba mejor con el planteamiento del problema. Por último se mostrará una evaluación sobre el modelo finalmente elegido e implementado.

3.2 Análisis de la situación actual

El Crawler es una herramienta diseñada para visitar los ficheros de sindicación de páginas Web y almacenar las nuevas entradas que se producen en cada sitio. De

este modo se va actualizando una base de datos con las nuevas informaciones disponibles en Internet.

Cuando se lanza el Crawler, se lee un fichero de entrada OPML. Este fichero puede estar definido en las propiedades del Crawler, o se puede definir como parámetro si se lanza el Crawler por línea de comandos. Una vez lanzado, los datos recuperados se almacenan en los ficheros de salida y también en la base de datos. Cuando el Crawler es lanzado, recorre todas las páginas que tiene definidas en el OPML como canal, añadiendo las nuevas entradas como nuevo registro.

El fichero OPML posée un parámetro que indica la fecha de la última actualización de la página a visitar. Este parámetro es útil para no descargar los últimos contenidos de la página Web si no se ha actualizado esta hora. Aún así, la página Web se visita para recuperar esta información.

El objeto del estudio es minimizar en la medida de lo posible el que se tenga que realizar esta comprobación. Con la ayuda de la estadística se va a diseñar una solución que modifique el fichero OPML de entrada para que el Crawler visite las páginas que se presupone que se han actualizado.

Se han estudiado algunos modelos de distribuciones de probabilidad para abordar el problema indicado. A continuación se muestran los análisis realizados para cada modelo y los inconvenientes encontrados en los mismos.

3.3 Definición del modelo

Uno de los aspectos a tener en cuenta a la hora de realizar el estudio, es que una página Web puede ser actualizada en cualquier momento a lo largo de un día. También hay que considerar que, a priori, no se puede saber si una página que va a visitar el Crawler estará actualizada. Es fácil adivinar que si el Crawler se está lanzando continuamente a lo largo de un día, por ejemplo cada hora, muchas veces se recuperarán páginas sin actualizar. En horarios nocturnos esta probabilidad aumentará considerablemente. La finalidad del estudio es, por tanto, establecer un modelo que indique con un alto grado de exactitud si una página va a estar actualizada cuando la vaya a visitar el Crawler. Más

concretamente, en el momento en que se va a lanzar el Crawler, saber qué páginas se deberían visitar.

La actualización de una página Web podría considerarse como un suceso de una variable aleatoria, que se produce de manera indeterminada a lo largo del tiempo. Y de acuerdo a esta proposición se ha realizado el análisis del modelo.

Modelo exponencial

Una variable que se da n veces de manera indeterminada (aleatoria) en una unidad de tiempo, sigue una distribución de Poisson. La cual consiste en la aparición de sucesos a lo largo de sucesivas unidades de medida. Por ejemplo, la llegada de clientes a la hora en cierto puesto de servicio, la aparición de defectos por área de superficie de material, llamadas de clientes de una centralita por minuto, etc. La probabilidad de que se de un suceso en una determinada unidad de medida es $P(X = r) = \frac{\lambda^r}{r!} e^{-\lambda}$; r=0, 1... Siendo X una variable de Poisson, en la que por término medio se observan λ sucesos por unidad de medida.

En el caso que se aborda, el número de actualizaciones diarias de una página podría ser esa variable aleatoria, por lo que este número se corresponderá con el parámetro λ de una distribución de Poisson, en la que la unidad de tiempo será la hora para nuestro estudio.

Pero en realidad, más que el número de veces que se actualiza cierta página diariamente, lo que interesa saber es cuándo se espera que vuelva a estar disponible una actualización de la misma. Para ello deberemos recurrir a otra distribución diferente en el enfoque, aunque de idénticas características: una distribución exponencial, la cual posee el mismo parámetro λ de la distribución de Poisson.

La función exponencial indica el tiempo transcurrido entre dos sucesos de una distribución de Poisson. Por ejemplo, si se evalúa la llegada de clientes a un cierto puesto de servicio, el número de clientes por hora será una variable de Poisson, pero el tiempo entre dos clientes consecutivos será una variable exponencial.

La probabilidad de que un suceso tarde en producirse más de una determinada unidad de medida se calcula como $P(T > t) = e^{-\lambda t}$, siendo el suceso contrario $P(T \leq t) = 1 - e^{-\lambda t}$.

Con la distribución exponencial se puede calcular precisamente el tiempo esperado hasta la siguiente ocurrencia. En el caso que se analiza, que se actualice la página Web.

Se realizará un ejemplo para entender con mayor claridad lo explicado anteriormente:

Los administradores de una página Web de noticias sobre política han realizado doce actualizaciones a lo largo del día añadiendo nuevo contenido. Con este dato, el valor del parámetro de Poisson es λ=0,5 (son 12 actualizaciones en un día, y considerando que la variable temporal será la hora, se tendrá $\lambda = \frac{12}{24}$*). Con estos datos se obtendrá un tiempo de actualización de* $\frac{1}{\lambda} = \frac{1}{0,5} = 2h$*, que es equivalente a decir que se actualiza una vez cada 2 horas (Se debería usar la función de probabilidad y no la esperanza, pero para entender el concepto es indiferente).*

Si el proceso realiza los cálculos de prioridades, por ejemplo, todos los días a la 01:00 h de la mañana, para que no haya demasiados servicios corriendo que puedan afectar al rendimiento del procesador, entonces se hallaría este valor de 2 horas, por lo que se especificaría al Crawler que ese día pasara cada dos horas por la Web para actualizar los cambios que debería tener.

Con estos datos, a las 03:00 h de la madrugada el Crawler visitaría la página Web, sin obtener ninguna actualización. A las 05:00 h volvería a pasar, obteniendo el mismo resultado. Así hasta las 09:00 h o incluso las 11:00 h, que serán horas más probables de que se vaya a realizar una nueva actualización.

Un gráfico que ilustre el ejemplo que se acaba de poner podría ser el siguiente:

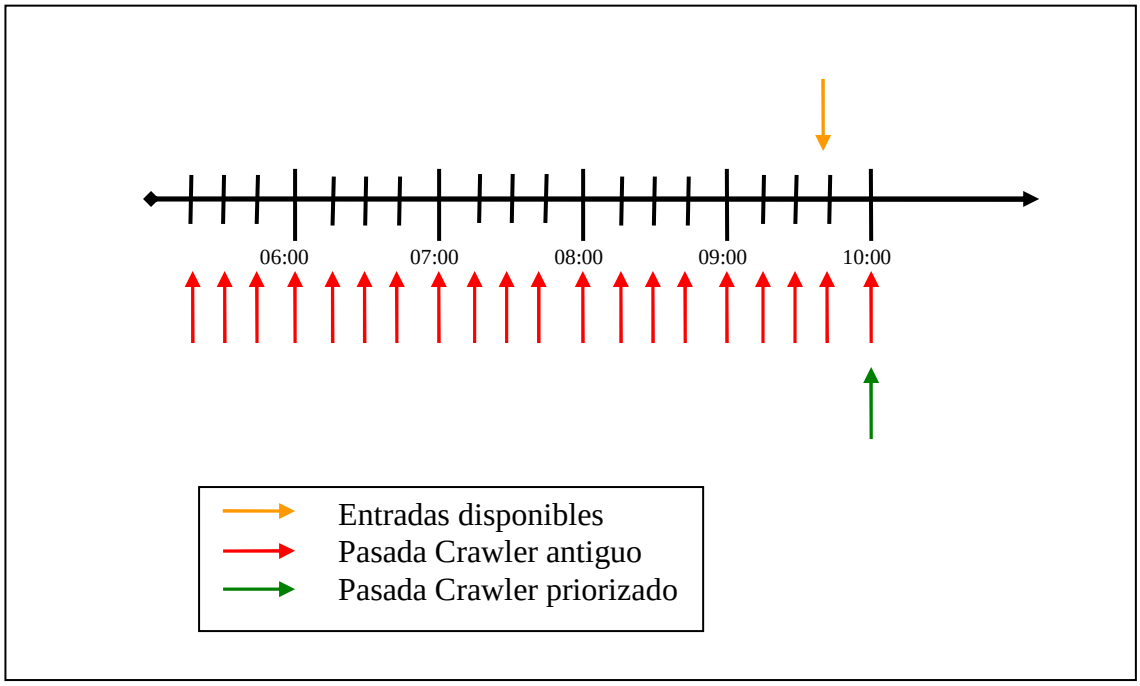

Figura 1. Figura representativa de la actualización en horario nocturno.

Con el ejemplo anterior se intenta mostrar la necesidad de conocer de algún modo la manera de actualizar de cada página. El horario, por así decirlo. Pero es evidente que no se puede utilizar la distribución exponencial, puesto que el tiempo medio de actualizaciones podría llevar a un horario no válido, como ha sido el caso. Es por ello que se ha enfocado de otra manera el procedimiento de actualización de las páginas Web.

Modelo de Poisson

Es evidente que cada persona (o empresa) actualiza su Web cuando puede. También es cierto que los horarios de actualización suelen ser más o menos homogéneos. Una empresa que se dedique a la publicación de noticias a través de Internet, tendrá un horario de trabajo durante el cual sus páginas se actualizarán con gran frecuencia, mientras que cuando termine ese horario será muy extraño encontrar actualizaciones. Un estudiante que tenga su propia Web, se dedicará a actualizarla al acabar sus horas lectivas, por lo que también seguirá un horario habitualmente.

Por lo general, en la mayoría de los portales de noticias o blogs se seguirán unos horarios particulares para cada página. Con lo cual se deberá tener en cuenta, además del número de veces que se actualiza un sitio Web, las horas más frecuentes de actualización del mismo.

Cada página tendrá unos horarios de actualización que se tendrán que calcular. Para ello se crearán ***rangos horarios***. Estos rangos serán definidos en la configuración de la herramienta, y podrán ser modificados según las características del campo al que se quiera aplicar. Un rango será válido si es divisor de 24, que son las horas que tiene un día. Para la explicación del modelo se mantendrá un rango de dos horas (r=2 a partir de ahora).

Dentro de un rango, se obtendrá el número de actualizaciones, y de este modo se podrá establecer una prioridad en la lectura del sitio Web. Si por ejemplo de 10–12 h se han añadido 20 contenidos nuevos a una Web, se deberá lanzar el Crawler para leerla más veces durante ese periodo que si se hubiera actualizado una sola vez. Por tanto, si el Crawler está configurado para que se lance cada 15 minutos, sería razonable que visitara la página de la primera opción las ocho veces que se lanzara en el periodo 10-12, puesto que es probable que en todas las visitas se recupere nuevo contenido. Mientras que para la segunda opción (una única actualización en r) se podría lanzar una sola vez, o incluso esperar a otro intervalo de tiempo. De cuándo se debería visitar la página se hablará más adelante.

Se va a desarrollar el mismo ejemplo mostrado anteriormente pero con el nuevo modelo aportado:

Se disponía de una página Web a la que se había actualizado doce veces a lo largo del día.

Sobre la 01:00 h de la madrugada se realiza el proceso que calcula los lanzamientos del Crawler para el día completo. La página ha sido actualizada con cuatro contenidos nuevos entre las 14:00-16:00 y con otros ocho contenidos entre las 20:00-22:00, puesto que son el tiempo libre de la comida y la salida del lugar de trabajo. Así que se configurará el Crawler para que visite la página

una vez entre las 14:00-16:00 y dos veces entre las 20:00-22:00, ya que lo más probable es que cuando pase por la Web recoja nuevos contenidos.

Se puede observar que la diferencia con el modelo anterior es considerable. En el primero se ejecuta el Crawler en periodos que no suele haber actividad en la mayoría de las páginas almacenadas en la base de datos, con lo que estamos consumiendo recursos sin ningún valor.

En el segundo modelo se acotan las posibilidades de error en el lanzamiento del Crawler, haciendo que las ejecuciones del mismo sean más eficientes puesto que pocas veces deberían producirse lecturas sin actualización. Además, en el segundo modelo no se tiene en cuenta el tiempo que pueda transcurrir hasta la siguiente actualización, sino el número de actualizaciones que se tiene de media en un rango r. Es decir, se toma como referencia la distribución de Poisson y no la distribución Exponencial.

Es por tanto un modelo más optimizado para las características requeridas.

Acotación

Se ha podido comprobar que es necesario dividir en rangos el día para calcular de manera más eficiente las próximas actualizaciones del Crawler, debido a los posibles horarios de los actualizadores. Se puede deducir, de manera similar, que si las actualizaciones diarias disponen de horarios, las actualizaciones semanales también los tendrán. Generalmente, los días laborales no tendrán los mismos horarios que los días festivos. Pero incluso entre sí, los días laborales también pueden ser diferentes. Un lunes tendrá (lógicamente) más similitud de horarios con el lunes de otra semana, que con un jueves. Es por esto que, además de rangos diarios (r), se tendrán ***agrupaciones*** de los días de la semana.

Entonces, según esta conjetura, un lunes de una semana cualquiera debería tener las actualizaciones de manera más o menos homogénea que los lunes de semanas anteriores o posteriores. Y con el resto de días de la semana pasaría lo mismo entre ellos.

Por lo tanto, para calcular la probabilidad de que en un lunes entre las 10:00 h y las 12:00 h haya actualizaciones, no se tendrán en cuenta las actualizaciones del día anterior, sino del lunes de la semana anterior. Y, de hecho, no solamente se tendrán en cuenta las actualizaciones del lunes de la semana pasada, sino de todos los lunes (hasta un extremo definido) de los que se tiene datos en el rango horario determinado.

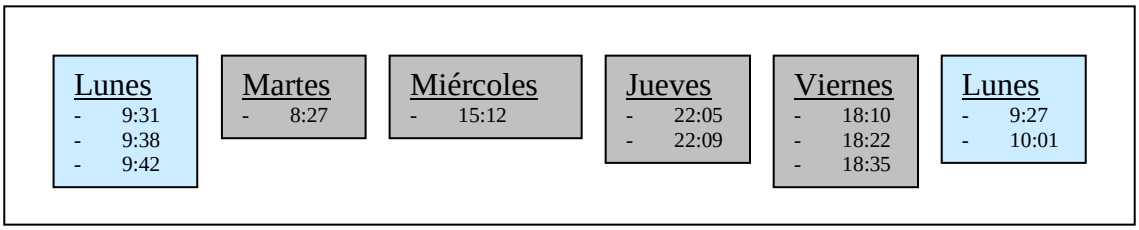

Figura 2. Figura representativa de la homogeneidad de actualizaciones según el día.

Continuando con el ejemplo de la página Web, se podría añadir lo siguiente:

Se sabe que los cuatro últimos lunes, se ha actualizado la página en el horario 10:00 h – 12:00 h con 3, 5, 3 y 4 entradas, respectivamente. Teniendo esto en cuenta, se obtendrá una media de $\frac{3+5+3+4}{4} = 4{,}5 = \lambda$ *. Es decir, que en el rango de 2 horas se tendrá de media de 4,5 actualizaciones. Si el Crawler se lanza cada 15 minutos, se podría configurar que cada dos lanzamientos visite la página Web.*

Conceptos destacados

Hasta ahora se ha definido el modelo como una serie de conceptos entrelazados:

- Se regirá mediante una distribución de Poisson.
- Estará dividido por rangos horarios.
- Se agrupará por días de la semana.

En ningún momento se ha determinado el tiempo entre lanzamientos del Crawler, ni el factor a tener en cuenta para que se visite o no una página al ser

lanzado. Este factor no es otra cosa que probabilidad, y se estimará a partir de los casos de prueba que se consideren para ajustar los valores necesarios.

En secciones posteriores se explicará la fórmula que permitirá realizar este cálculo.

Con respecto al tiempo entre lanzamientos del Crawler, puede resultar intuitivo que debe ser un divisor del rango horario elegido. Pues así se podrá dividir en partes iguales el tiempo entre lanzamientos. En los ejemplos anteriores se ha estado usando un r = 2 h y un t=15 min. Teniendo en cuenta que 2 h = 120 min, se puede observar que el tiempo se divide en 8 partes iguales. Por cada rango horario, el Crawler se lanzaría 8 veces.

Combinación lineal

Se ha explicado que se va a seguir como referencia las actualizaciones del mismo día de la semana para la predicción de actualización. Pero ¿de cuántos días se deberán tener datos? Únicamente los datos de la semana anterior podrían resultar poco fiables, puesto que es probable que un martes se actualicen muchas entradas debido a un suceso aislado, y que al siguiente martes no se actualicen apenas. En tal caso se podría caer en el error de que los datos sean demasiado heterogéneos entre sí, al no llevar un patrón similar al habitual. Sería posible, por ejemplo, que un martes hubiera sido fiesta y, al estar en casa, se hubieran actualizado 10 entradas. Con este dato, para el martes siguiente se predeciría esta media de 10 entradas en el rango dado. Sin embargo, al haber sido día laboral, se ha podido actualizar una sola entrada, pero con la configuración calculada, el Crawler ha visitado la página varias veces sin obtener éxito.

Pero entonces, ¿dónde habría que poner el límite? ¿Podría ser un mes? Con un mes se podrían tener datos suficientes. Pero también es cierto que un mes se pueden tener horarios particulares (vacaciones, proyectos nuevos, jornada intensiva, etc.). Entonces ¿cuál sería el punto exacto? ¿Quizás seis meses? ¿Habría que disponer de todos los datos disponibles en la base de datos?

Cabe pensar que a partir de seis meses, la estimación de entradas que se haga sea lo más acertada posible. Sería intuitivo pensar que cuantos más datos tengamos

en cuenta, mayor será la fiabilidad de la estimación. Sin embargo, en la realidad esto no es así. La mayor parte de las páginas tarde o temprano dejan de actualizarse. O lo hacen pero no con el mismo ritmo al que lo hacían. De acuerdo a los blogs es más común que haya altibajos en el tiempo. La gente deja de actualizarlos o va reduciendo su constancia. Sobre todo para los que poseen algún blog a modo de hobbie. El auge de las redes sociales ha hecho que se pase más tiempo en estos portales y que se actualicen tanto o más estas redes que los propios blogs. Por lo tanto, es razonable no utilizar todos los datos de que se puedan disponer en la base de datos para la estimación de probabilidades de visita, sino usar datos entre seis meses y un año.

En esta última opción, también habrá fechas en que los datos no sean siempre igual de homogéneos. Ya sea por vacaciones, más trabajo de lo habitual o cualquier otra causa.

El error más común que se puede cometer es si hay gran diferencia en el número de actualizaciones a largo plazo.

Es posible que al principio de los seis meses una Web apenas se actualizara. Por ejemplo seis actualizaciones en los cinco primeros meses (dentro de un rango específico y en un día de la semana concreto) ya que se está empezando y hay más tareas que atender. Pero en el último mes la ocupación es menor y se saca contenido de muchas partes, por lo que se actualiza hasta treinta veces la Web. Con estos cálculos, se tendrían 36 actualizaciones en 6 meses, (se contará un día por cada semana puesto que se cuentan los homónimos) que teniendo en cuenta que cada mes tendrá 4 días de la semana, se obtendría $\frac{36}{24} = 1{,}5$, que sería el valor de la variable λ. Como se tiene un valor de una vez y media de actualizaciones al día, el Crawler la visitaría una única vez. Sin embargo, si en la siguiente semana se ha seguido el ritmo del mes anterior, debería haber aproximadamente 7 actualizaciones, que dependiendo de cuándo pasara el Crawler, podrían quedarse sin recuperar.

Así que es difícil afinar un límite para recuperar las actualizaciones y obtener la variable λ. Pero si se tuviera en cuenta estas tres opciones y se relacionaran, se podría obtener un valor más aproximado al exacto. Se podría hacer una

combinación lineal de las tres prioridades calculadas. Cada una de ellas sería una variable aleatoria de Poisson por sí misma, y la combinación lineal también lo sería.

Así que se usará la siguiente fórmula.

$\lambda = \alpha w + \beta m + \delta h$, en donde:

- w es la variable aleatoria de Poisson calculada con los datos para el mismo día de la semana anterior (day of **w**eek)
- m es la variable aleatoria de Poisson calculada con los datos para el mismo día de todas las semanas del mes anterior (**m**onth)
- h es la variable aleatoria de Poisson calculada con los datos para el mismo día de todas las semanas de los seis meses anteriores (**h**alf a year)
- α, β y δ son los coeficientes asociados a estas variables, que se fijarán en principio como 0,2; 0,3; 0,5 respectivamente, considerando que será más preciso cuanto más tiempo se abarque. Estas variables se podrán modificar a lo largo del estudio, teniendo siempre en cuenta que α + β + δ = 1.
- Como w, m y h eran variables aleatorias de una distribución de Poisson, la combinación lineal de ellas también será una variable aleatoria de Poisson, y por tanto tendrá las mismas propiedades.

Si se consideran los datos anteriores que se usaban como ejemplo, la última semana se hicieron 8 actualizaciones (no se dijo nada sobre la semana pero lo asignamos ahora), el último mes se hicieron 30 y el último semestre 36. Con lo que se tendrá $w = 8$, $m = \frac{30}{4} = 7{,}5$ y $h = \frac{36}{24} = 1{,}5$.

El cálculo da $\lambda = \alpha w + \beta m + \delta h = 0{,}2 * 8 + 0{,}3 * 7{,}5 + 0{,}5 * 1{,}5 = 4{,}6$ actualizaciones en el periodo de 10:00 – 12:00 de un martes, por los 1,5 que salía anteriormente calculando sólo el semestre.

3.4 Funcionamiento del modelo

Se va a definir un requisito para el funcionamiento tel modelo. Para que el Crawler visite una cierta página, será necesario que la variable λ tenga un valor, como mínimo, de 1. Con ello se quiere asegurar que cuando se visita una página, sea porque se espera que al menos haya una actualización. Tampoco se asegura que esa actualización vaya a estar, pero el considerar un valor menor a este restaría eficiencia al proceso.

El que λ tenga como cota mínima el valor de 1, implica que se puede calcular también un umbral mínimo para la probabilidad. El valor de este umbral será el resultado de calcular la probabilidad de que haya al menos una visita con el parámetro $\lambda = 1$. El resultado es, por tanto:

$$P(X > 0) = 1 - P(X \leq 0) = 1 - \frac{\lambda^r}{r!} e^{-\lambda} = 1 - \frac{1^0}{0!} e^{-1} = 0{,}6321$$

A este valor lo llamaremos ***Cota de probabilidad***. Si la probabilidad de que una página se visite es igual o superior a este umbral, el Crawler la visitará.

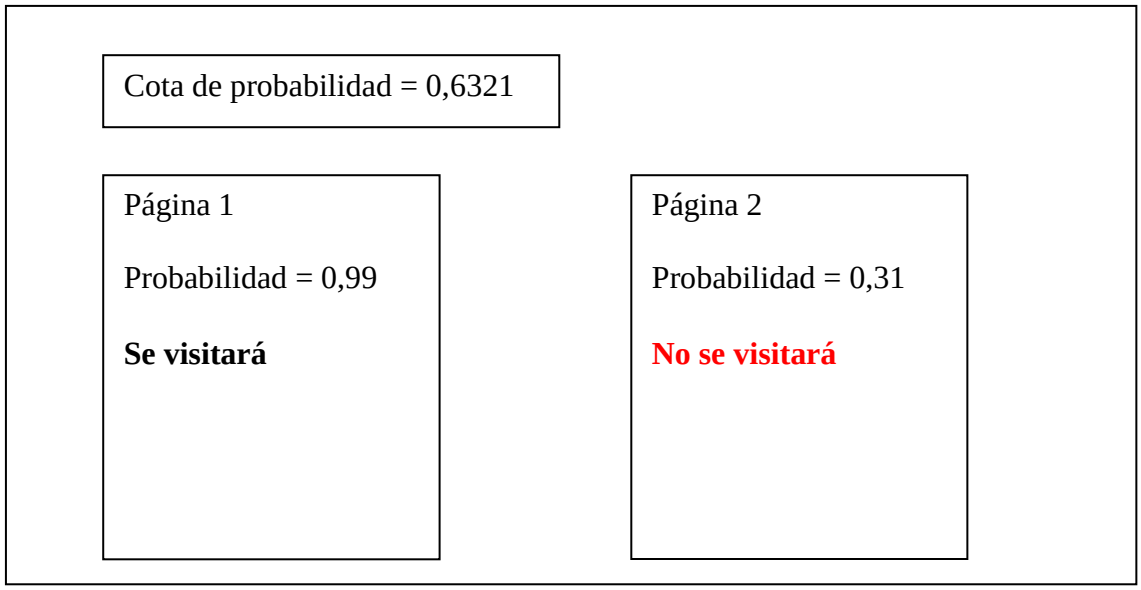

Figura 3. Figura representativa del uso de la cota de probabilidad.

En el ejemplo con λ=4,6 actualizaciones en un rango, se obtiene que la probabilidad de que se visite la página es $P(X>0)=1-P(X\leq 0)=1-\frac{4,6^0}{0!}e^{-4,6}$. El resultado es $P(X>0)=1-0,01=0,99$, que es un valor superior a 0,6321, y por lo tanto se visitaría la página.

Con un valor para λ de 4,6, se puede pensar que la página tenga entre 4 y 5 actualizaciones en el rango indicado. Si se halla la probabilidad de que se realicen, por ejemplo, más de 5 visitas en ese rango, se obtendría $P(X>5)=1-P(X\leq 5)=1-0,69=0,31$. Este valor es inferior, por lo que no se visitaría de nuevo la página si se hubieran recuperado ya 5 entradas.

Problemas

Ahora ya se sabe cuando visitará una determinada página el Crawler. El inconveniente que hay en este método, es que no se sabe en qué momento dentro del rango se actualizará la página. Haciendo simplemente el cálculo, se podría

asociar este resultado a que en el rango se visite la página cada vez que se lance el Crawler hasta que se consigan al menos 5 actualizaciones.

El funcionamiento no se realizará de este modo. Lo que se quiere es que el Crawler tenga que visitar el menor número de veces posible la página, pero a la vez recuperar todas las visitas. Por lo tanto, se debería intentar lanzar el Crawler lo más tarde posible.

En este ejemplo se actualiza menos de cinco veces de media, pero según el cálculo de probabilidad se tendría que visitar la página, puesto que se ha obtenido un $\lambda=4,6$, y es posible que estas actualizaciones se hagan al final. Es obvio que mientras que no se recojan actualizaciones, se visitaría en todas las pasadas, debido a que la probabilidad sería siempre la misma.

Una variante que corrige este defecto es calcular la probabilidad de que se actualicen tantas veces como las pasadas que le quedan por hacer al Crawler dentro del rango. Esto es, que si el Crawler está configurado como t=15', y por tanto se lanza 8 veces en un rango, primero se calcularía $P(X>7)$ –con 8 pasadas se calcularán actualizaciones de 0 a 7-, que en este caso concreto daría un resultado menor a 0,65 y no se pasaría por la página. En la siguiente pasada, se calcularía $P(X>6)$. Y se seguiría descendiendo con cada pasada del Crawler hasta llegar a $P(X>0)$, es decir, que tenga al menos una actualización. En este punto se observaría que en la última pasada del Crawler, si la página se tiene que visitar una vez, se visitará aquí. Por lo tanto se espera que se haya actualizado ya porque el rango se habrá completado.

Otro problema que puede surgir es que se recojan muchas actualizaciones al inicio del rango. Si, por ejemplo, $\lambda=10$, Para $P(X>7) = 0,78$ se visitaría la página. Si se recuperaran 8 entradas porque se han añadido al inicio del rango, la siguiente pasada, para $P(X>6)$ el resultado seguiría siendo mayor que 0,65 y se seguiría visitando la página. Sin embargo es evidente que la probabilidad habría bajado. En el resto de pasadas también se realizaría visita.

Por tanto, hay un dato más a tener en cuenta: las actualizaciones recogidas.

Se denotarán por la letra A, y se sumarán a las pasadas del Crawler, de manera que para calcular la probabilidad se necesite saber:

P(X > t + A), siendo t el número de pasadas que le quedan por hacer al Crawler en un rango, y A el número de actualizaciones recogidas en ese rango.

Con esta fórmula, teniendo λ = 10, en la primera pasada se hallaría P(X > 7 + 0, que resultaría lo mismo que antes, 0,78. Sin embargo, al recogerse 8 actualizaciones, la siguiente pasada se tendría P(X > 6 + 8) = P(X>14), que daría un valor inferior a la Cota de probabilidad. Así se llegaría al final de pasadas, en donde t = 0 y P(X > 8), que en este caso daba el 0,78 de nuevo, y se intentaría recoger el resto de actualizaciones, si las hubiere.

3.5 Estructura de datos

Para realizar la mejora del proceso de priorización en las visitas a las webs, se necesitará disponer de una serie de datos almacenados para poder realizar el estudio previo y calcular las visitas del Crawler. Para ello se tendrá que disponer de los siguientes datos representados en tablas:

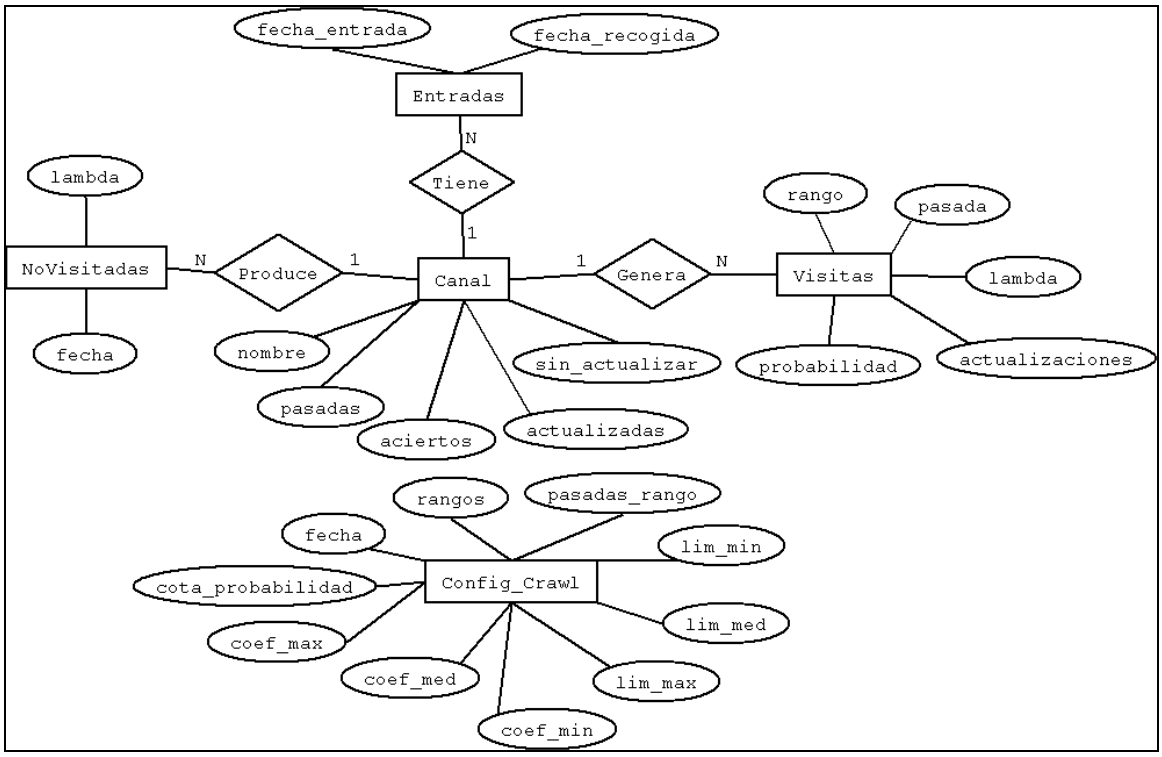

Figura 4. Esquema relacional de la base de datos.

Cuya descripción es la siguiente:

- Entradas (canal, fecha_entrada, fecha_recogida), con los datos relativos a las entradas recogidas.
    - Canal → Una página web unívoca
    - Fecha_entrada → fecha y hora de actualización de la entrada en la página web
    - Fecha_recogida → Fecha y hora de recogida de la entrada por el Crawler
- Visitas (canal, rango, pasada, lambda, actualizaciones, probabilidad), con los datos relativos a las futuras pasadas del Crawler para un rango.
    - Canal → Una página web unívoca
    - Rango → el periodo en el que se deben realizar las pasadas
    - Pasada → el número de pasada dentro del rango en el que el Crawler visitará la página

  - Lambda → resultado del cálculo del parámetro de Poisson para las entradas de un canal en un rango

  - Actualizaciones → número de actualizaciones en las pasadas actuales del rango

  - Probabilidad → posibilidad de que se visite la página en la siguiente pasada del Crawler. Un valor entre 0 y 1, deberá ser mayor que la Cota de probabilidad para que se realice la visita.

- Config_Crawl (fecha, rangos, pasadas_rango, lim_min, lim_med, lim_max, coef_min, coef_med, coef_max, cota_probab), con los datos de configuración del Crawler.

  - Fecha → para guardar un histórico de las configuraciones y utilizar siempre la última. Para el cálculo se tendrá en consideración la última fecha guardada hasta el día anterior.

  - Rangos → número de rangos en un día (r). Ha de ser divisor de 24 horas. Por defecto se utilizará r = 12.

  - Pasadas_rango → número de pasadas en un rango (p). Deberá ser divisor de $\frac{24 \times 60}{r}$. Con el valor por defecto de r deberá ser divisor de 120 minutos. Su valor por defecto será p = 8, que resultará una pasada cada 15 minutos.

  - Lim_min → número de semanas para el cálculo de w. Por defecto 1 semana.

  - Lim_med → número de semanas para el cálculo de m. Por defecto 4 semanas.

  - Lim_max → número de semanas para el cálculo de h. Por defecto 24 semanas.

  - Coef_min → Coeficiente de w estimado para la ponderación de la ecuación principal. Coef_min = α. Por defecto 0,2.

  - Coef_med → Coeficiente de m estimado para la ponderación de la ecuación principal. Coef_med = β. Por defecto 0,3.

  - Coef_max → Coeficiente de h estimado para la ponderación de la ecuación principal. Coef_max = δ. Por defecto 0,5.

- NoVisitadas (canal, fecha, lambda), con los datos relativos a las páginas que no se van a visitar con el Crawler. O bien por tener un λ menor que 1, o porque no tienen siquiera actualizaciones en el periodo h.

  - Canal → Una página web unívoca

  - Fecha → Fecha de la última visita. Si es muy lejana se podrá optar por no visitarla ni siquiera en horarios poco ociosos (fuera del modelo).

  - Lambda → resultado del cálculo de las entradas de un canal en un rango. Puede ser nulo si no tiene visitas.

- Canal (nombre, pasadas, aciertos, sin_actualizar), con los datos relativos a los canales que se visitan.

  - Nombre → Es la dirección del canal que se visita.

  - Pasadas → Es el número de pasadas que se han realizado para este canal. Junto con los aciertos se puede establecer la precisión.

  - Aciertos → Es el número de veces que se ha tenido éxito al realizar pasadas. Junto con las pasadas se puede establecer la precisión.

  - Actualizadas → Es el número de entradas que se han actualizado en su rango correspondiente. Junto con sin_actualizar se puede establecer la cobertura.

  - Sin_actualizar → Es el número de entradas que se han quedado sin actualizar en su rango correspondiente. Junto con actualizadas se puede establecer la cobertura.

3.6 Cálculo de errores

En el estudio del comportamiento del modelo se deberá hacer hincapié en el cálculo de errores de la estimación. Para ello se analizarán dos tipos diferentes de errores:

- Error de precisión → $\mathbf{e_p}$ será la visita del Crawler a una página sin recuperar actualizaciones. Se incrementará en una unidad por cada visita infructuosa.

- Error de cobertura → $\mathbf{e_c}$ será las entradas que no se recuperan porque no se ha lanzado el Crawler. Se incrementará en el número total de entradas que se debían haber recuperado.

A su vez, se considerarán también dos tipos diferentes de aciertos. Se denotará por **s** (success):

- Acierto de precisión → $\mathbf{s_p}$ será la visita del Crawler a una página recuperando actualizaciones. Se incrementará en una unidad por cada visita fructuosa.

- Acierto de cobertura → $\mathbf{s_c}$ será el total de entradas recuperadas al lanzarse el Crawler. Se incrementará en el número total de entradas recuperadas.

Para el modelo anterior, en el que se realizan todas las pasadas en cada rango, $\mathbf{e_p}$ y $\mathbf{s_p}$ son elementos opuestos. El total de pasadas podría expresarse como $\mathbf{e_p + s_p}$.

Con estas variables se podrán calcular otras dos muy importantes:

- Precisión → será la relación entre los aciertos y los errores de precisión. Se calculará mediante la fórmula $precisión = \frac{s_p}{s_p + e_p}$, y representará el porcentaje de visitas fructuosas en el total de visitas.

- Cobertura → será la relación entre los aciertos y los errores de cobertura. Se calculará mediante la fórmula $cobertura = \frac{s_c}{s_c + e_c}$, y representará el porcentaje de entradas recuperadas del total de entradas disponibles.

Hay que destacar que la precisión se puede establecer a partir de una pasada del Crawler, puesto que se sabe si se ha visitado una página y el resultado ha sido o no fructuoso. Sin embargo, la cobertura se tiene que establecer *a posteriori*, puesto que hasta una nueva pasada a una página con entradas sin recuperar, no se podrá saber que estas entradas se habían quedado en el aire.

3.7 Línea base

Teniendo en cuenta que en el modelo anterior, el Crawler visitaba en cada pasada todas las páginas, la comparación de dicho modelo con el Crawler priorizado debería resultar en que el error de precisión cometido por el modelo anterior debería ser mayor que el error de precisión cometido por el Crawler priorizado. De modo que se debería obtener un resultado de **e_p Priorizado < e_p Actual**.

Según los datos estudiados en la simulación del modelo que se explica más adelante, se han obteneido unos resultados de media para el total de diez ensayos:

- Crawler actual:
  - Precisión → P = 0,1435
  - Cobertura → C = 1
  - Media ponderada → M = 0,4004
  - Pasadas → 992
- Crawler priorizado:
  - Precisión → P = 0,7203
  - Cobertura → C = 0,7369
  - Media ponderada → M = 0,7253
  - Pasadas → 60

Aunque el valor óptimo sería el de conseguir una precisión cercana a 1.

La comparación del error de cobertura carece de sentido, puesto que en el modelo anterior, al visitar en todas las pasadas, nunca deja entradas sin actualizar en un rango.

A su vez, se debe tener en cuenta que el número de aciertos debe ser mayor que el número de errores de precisión. De modo que **s_p Priorizado - e_p Priorizado > 0.** Mientras que con el modelo antiguo, **s_p Anterior - e_p Anterior < 0.**

El objetivo óptimo sería **e_p Priorizado ≈ 0** y **e_p Priorizado << e_p Anterior**.

Mediante las variables precisión y cobertura, se calculará una última, la **media ponderada**. Que resultará del cálculo de:

$$Media = 0{,}7 * precisión + 0{,}3 * cobertura$$

Como se puede apreciar en la fórmula anterior, se ha concluido que es más importante un error de precisión que un error de cobertura. Esto es debido a que los errores de precisión son accesos perdidos siempre, mientras que los de cobertura pueden ser soliviantados en otra pasada contigua. Para cálculos con probabilidades muy bajas en un rango, es posible que quede alguna entrada sin recuperar puesto que no se ha visitado, pero también es muy probable que se recoja en otro rango en el que la probabilidad sea mayor y, por lo tanto, haya visita.

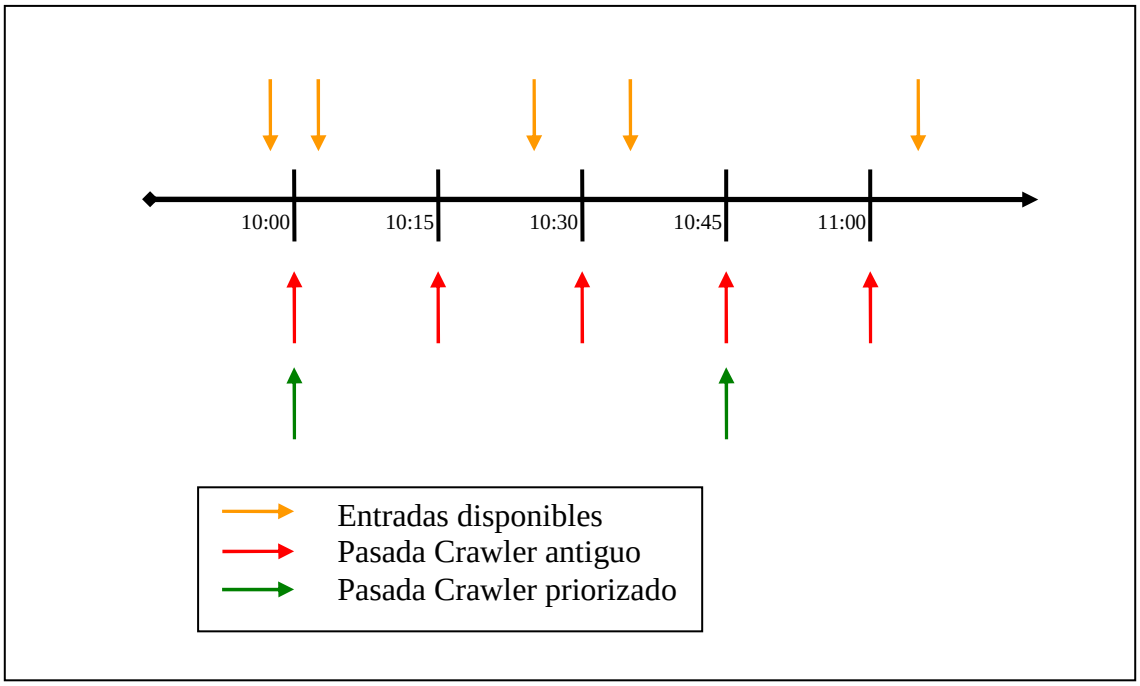

Figura 5. Figura representativa del uso de la cota de probabilidad.

En la figura anterior se ha representado la vida de un canal en un horario concreto. El canal dispone de 5 entradas a lo largo del horario especificado, repartidas a su vez en horas distintas. Se puede apreciar que el Crawler antiguo pasa cada quince minutos, mientras que el Crawler priorizado pasa en dos ocasiones. Para entender lo que es un error de cobertura basta con observar que desde las 10:00h en que pasa el Crawler priorizado, hasta las 10:45h que vuelve a pasar, ha habido tres actualizaciones en el canal, de las que dos han sido en rangos anteriores. Estas dos actualizaciones serían los errores de cobertura, puesto que había entradas para recuperar pero no se ha pasado el Crawler. El error de precisión se puede explicar observando en este caso el Crawler antiguo, que a las 11:00h realiza una pasada sin que haya actualizaciones, debido a que se recogieron todas las existentes en el rango anterior.

Con los valores que se obtengan del número de pasadas, precisión, cobertura y la media ponderada, se estimarán los valores ajustados de los coeficientes α, β y δ. Los valores iniciales serán de 0,2, 0,3 y 0,5 respectivamente, pero se irán ajustando según sean óptimas estas variables.

El ajuste de los coeficientes se explica con la simulación del modelo, que se ha realizado sobre un libro Excel.

3.8 Evaluación del modelo

El actual sistema que se encuentra en funcionamiento fuerza la visita de todas las páginas que hay almacenadas en la base de datos cada cierto tiempo. De este modo, la mayor parte de las veces que se lanza el Crawler no se lee nada, o se añaden muy pocos contenidos para la cantidad de páginas que se visitan.

Con el nuevo modelo para priorizar las lecturas, se consigue un comportamiento más eficiente con respecto a este problema, ya que solamente se leerá cuando se espera que haya contenidos. Así, además de evitar lanzar el programa cuando se sabe a priori que no se va a leer nada, se logra aproximar la hora de lectura a la hora de la última actualización de la página.

Uno de los problemas que se pueden encontrar con este modelo es que habrá páginas que no actualicen demasiado o que lo hagan en distintos rangos, de modo que tengan pocas actualizaciones pero repartidas a lo largo del día. Entonces es posible que no se llegue a planificar la lectura de estas páginas debido a la "inactividad" que tendrían.

Además, también podría darse el caso de que una página actualice demasiado. Que tenga tantas actualizaciones en un rango que las pasadas del Crawler no sean las suficientes como para recuperar todas esas actualizaciones antes de que hayan sido eliminadas del RSS. Aunque esto también puede suceder de la misma manera con el modelo actual, ya que para corregirlo sería necesario bajar el tiempo de espera hasta la siguiente ejecución del programa, que ahora mismo se encuentra en 15 minutos. Es decir, aumentar el número de pasadas del Crawler (p).

Capítulo 4

Simulación del modelo

4.1 Introducción

Se ha plasmado en un libro Excel lo que sería la funcionalidad del modelo descrita en este documento. Dentro del libro se ha simulado el proceso de generación de las visitas y recuperación de entradas.

El documento puede verse en este enlace:


Modelo Priorizado

Para esta simulación se ha considerado que todos los datos pertenecen al mismo día de la semana. Además, sólo se han realizado cálculos en cuatro rangos horarios. Los rangos están comprendidos entre las 8:00h y las 16:00h. Cada rango son 2 horas. El Crawler pasa 8 veces por cada rango, lo que equivale a decir que se pasa cada 15 minutos. Los límites serán de 1, 4 y 24 semanas, para el mínimo, medio y máximo respectivamente.

4.2 Entrada de datos

La primera hoja que aparece es “Datos”. En ella se realiza el cálculo de lo que serían las entradas de un canal. Se han simulado los datos relativos a 100 días. Con ello se tendrá la ocasión de empezar a realizar los cálculos a partir del día 25, para que se recojan las entradas aportadas en esos 6 meses (24 semanas).

La primera columna que se observa, bajo el título **Actualizaciones**, posee el número de actualizaciones que se harán en cada uno de los 100 días disponibles. El número de actualizaciones de cada día se ha calculado mediante el método Monte-Carlo, que calcula valores de una distribución de Poisson para un parámetro λ determinado. En este caso λ=5. La función se encuentra en una macro dentro del libro Excel. Si se quisiera recalcular el número de actualizaciones de un día, bastará con introducir el foco dentro de la celda (pulsando F2 por ejemplo), y a continuación pulsar Intro. Se puede observar que se recalcula el valor.

A continuación de la columna de actualizaciones se encuentran una serie de columnas con cada hora de actualización en un día. Hay 11 columnas porque no se ha dado el caso de que salgan más de 11 actualizaciones haciendo las simulaciones, pero en tal caso habría que añadir más columnas.

Cada celda perteneciente a la hora de actualización se compone de una función de distribución Normal, con media 11 y desviación típica 1. Esto es para que se centre en las 11:00 h y de resultados en el rango 10:00 h – 12:00 h a ser posible. Las celdas también tienen una condición asociada al número de actualizaciones, de manera que si el número es menor que su orden (del 1 al 11) no se genera el valor.

Para recalcular los valores de las horas en que se actualizan las páginas, basta con pulsar la tecla F9.

Por conflictos de formato a la hora de tratar los datos, se han volcado las horas generadas en esta hoja a la llamada “Hoja1” por medio de un copiado a un editor de texto y del editor copiado a dicha hoja. De manera que dentro del libro, se trabaja con los datos aportados en “Hoja1”. Es posible que no coincidan los datos

de las dos hojas, puesto que cuando se pulsa F9 varían los datos de “Datos”, pero no los de “Hoja1”. Si se quiere simular con datos distintos a los de “Hoja1” se tendrá que hacer el copiado mediante otro editor de texto (notepad) como se ha dicho anteriormente.

La segunda y tercera hojas, “Dia1” y “Dia2”, son las más importantes y son prácticamente idénticas. Si se da un vistazo por algunas celdas se puede observar que las operaciones son con referencias a otras celdas, con lo que para obtener resultados en sucesivos días bastará con copiar íntegramente la hoja “Dia2” y cambiar el dato relativo al día. Se explicará por tanto el contenido de la hoja “Dia2”.

Hay que tener en cuenta que cada vez que se quiera recalcular un dato hay que pulsar F9.

4.3 Hoja de plantilla

Se refiere a la hoja con nombre “Dia2”, que será la que haga las veces de plantilla para el resto de días.

En la parte superior izquierda se puede ver los valores referentes a la cota de probabilidad y a los coeficientes de la función principal del modelo. Estos valores están referenciados a la hoja “Resultados”, y si se cambian los valores en esa hoja y se pulsa F9, aparecerá reflejado el cambio en todas las hojas que se copiaron de “Dia2”.

En la parte superior central se encuentra el número de día del cálculo. Este número coincide con el número de hoja. El número de registro es el registro del que se obtienen los datos en “Hoja1”. Como se puede apreciar se comienza por el registro 25, que es el día siguiente al cálculo completo del semestre.

En la parte derecha de la hoja se encuentran algunas referencias para el cálculo de las variables de la función principal.

- w se refiere a los datos de la hoja “Hoja1” que se corresponden con las actualizaciones de la semana anterior. En este caso a la fila 25 (se está hablando de la hoja “Dia2”, si fuera “Dia1” tendría que ser la fila 24), desde la A hasta la K.

- m se refiere a los datos de la hoja “Hoja1” que se corresponden con las actualizaciones de 4 semanas atrás, de la 22 a la 25.

- h se refiere a los datos de la hoja “Hoja1” que se corresponden con las actualizaciones de 24 semanas atrás, de la 2 a la 25.

Estas celdas de referencia, hacen a su vez referencia a la celda E2, en donde se encuentra el número de día, para sumarlo siempre al día anterior. Así cuando se copie esta hoja en otra y se cambie de día, se tendrá la actualización automáticamente.

Lo siguiente que se va a observar es el recuadro grande que ocupa la mayor parte de la pantalla y que está unido a la columna numérica de Excel. Hay 4 recuadros como este en cada hoja. Se corresponden a los 4 rangos en los que se va a realizar la simulación y se explicará el funcionamiento de uno, ya que el del resto es idéntico, pero en su propio rango.

4.4 Cuadro principal

En la parte superior se encuentran las horas de pasada del Crawler dentro del rango. Puede observarse que son 8 pasadas. Y que cada pasada se realiza en el último minuto del periodo. El siguiente minuto correspondería a otro periodo y puede que incluso a otro rango.

Dentro del recuadro, en la parte superior izquierda se encuentra la duración del rango, con la hora inicial, y la hora final. Estos datos son solo visuales.

Debajo se puede observar una pequeña tabla. En la parte de la izquierda están los nombres de las variables de la fórmula principal del modelo. Y en la parte derecha están los valores de estas variables.

El valor de lambda se calcula precisamente con la fórmula principal. Si se mira detalladamente la fórmula se podrá observar cómo las celdas correspondientes a los coeficientes (que estaban al comienzo de la hoja) son fijas. Tienen $ para que no se modifiquen si se arrastra o copia la celda. Esto es así para que resulte sencillo copiar los recuadros dentro de la misma hoja, ya que los valores de las variables se calculan por cada rango, pero los valores de los coeficientes son fijos. De este modo se podrían añadir más recuadros con rangos a la hoja sin tener que rellenar muchos datos.

Los valores de las variables se calculan con una función algo difícil de ver en Excel, pero que en el fondo es sencilla. Lo que en definitiva se consigue con cada función es recuperar los valores de la semana, mes o semestre anterior.

Para ello se usan dos funciones predeterminadas de Miscrosoft Excel: Contar.Si e Indirecto.

4.5 Funciones destacadas

Contar.SI

La función Contar.Si comprueba si en un rango se cumple una determinada condición y, en caso de cumplirse, suma una unidad por cada ocurrencia. O lo que es lo mismo, si hay 10 entradas en este rango, se devolverá 10. Los tres valores se dividen después por el número de semanas que abarcan, para obtener la media con respecto a un mismo día, ya que se tiene que trabajar con las mismas magnitudes.

Indirecto

La función Indirecto lo que hace es recuperar el contenido de la celda a la que se referencia. En este caso, INDIRECTO(L3) devuelve el contenido de la referencia a la que apunta L3. Es decir, que devolvería el contenido de Hoja1!A25:K25, que se corresponde con el rango de la semana anterior. Esta

función se usa porque en la hoja se repiten mucho estas referencias. Y en caso de alguna equivocación es más sencillo modificar la celda en la que está la referencia que toda la hoja.

4.6 Obtener entradas

Siguiendo con la función que obtiene los valores de las variables, lo que obtiene finalmente son las entradas totales que hay en el día (condición >=0:00). Y luego se restan las que son menores que el inicio del rango (condición <8:00) y las que son mayores que el fin del rango (condición >9:59). Así, lo que se ha hecho ha sido quedarse con las entradas que tengan su hora entre las 8:00 y las 9:59, ambos inclusive.

Las tres funciones calculan lo mismo, pero cada una para su rango.

Por ultimo se debe restar un último valor a cada función (antes de realizar la división por el número de semanas), que es el número de entradas que aún no se actualizaron. En la aplicación, este valor no tiene sentido, pues las entradas que no se han actualizado no se encuentran en la base de datos y no se usan para la generación de visitas. Pero en el Excel ha de hacerse puesto que las entradas se encuentran escritas en la hoja “Hoja1” y es la única manera de comprobar si se habían recuperado ya.

El valor se calcula con respecto a la hoja anterior únicamente. Y el cálculo se encuentra a la derecha de la tabla de las variables. Para ello se comprueba que no haya habido actualizaciones ni errores de precisión en los rangos del día anterior en orden descendente hasta llegar al mismo rango, sin incluirlo. En caso de no haber ninguna de las dos cosas, se sumarían el número de entradas que se quedaron sin recoger en el mismo rango en que se está, pero del día anterior. Que es posible que fueran 0 también.

Se comprueba si ha habido actualizaciones o errores de precisión porque en cualquiera de los dos casos se ha visitado el canal, por lo que al visitarse se han

tenido que descargar los datos que fueran del mismo rango en el que se está. Y esos datos ya los habría calculado en la función.

4.7 Variables internas de cálculo

Una vez explicado el cálculo de las variables, se van a ir explicando las restantes filas por el título de cada una de ellas.

Pasada en rango: Es el número de la pasada del Crawler dentro del rango. Como hay 8 pasadas va de 0 a 7.

Pasa: Indica con un SI cuando la probabilidad de que el Crawler visite la página en el rango es mayor que la cota de probabilidad y con un NO cuando no lo es. La probabilidad está en la celda a la que hace referencia.

Disponibles: Comprueba si en el periodo estimado hay entradas disponibles. De ser así las suma. Para ello se utiliza la misma función que para hallar el valor de las variables, pero modificando las condiciones para calcular en el periodo deseado.

Recupera: Comprueba si se visita la página y, en caso de hacerlo, suma las entradas que hubiera disponibles. Si se ha visitado la página, se hace la comprobación de los periodos anteriores para sumar las entradas que hubiera disponibles. Se comprueban los periodos anteriores siempre que no se haya visitado la página en los mismos, ya que de este modo ya se habrían recuperado y lo se tendría anotado.

Error precisión: Comprueba si se ha visitado la página. En caso de visitarse comprueba si se han recuperado entradas. Si la primera condición es verdadera y la segunda es falsa, se muestra SI para confirmar que ha habido un error de precisión, que se ha visitado la página sin recuperar datos. En cualquier otro caso se muestra NO.

Error cobertura: Si no se ha visitado la página y había entradas disponibles se marca el número de entradas sin recuperar aquí.

P(X>...): Es el valor de t+A en P(X>t+A). En principio es la diferencia de 7 menos el número de pasada actual. Pero si se han recogido entradas se suman a este valor.

Probabilidad: Función de probabilidad de Poisson que se calcula con el valor calculado en P(X>...) y el λ calculado.

Todos los valores anteriores se calculan en cada periodo dentro del rango. Los de debajo serían los totales por rango.

Sin actualizar pasada: Son todas las entradas que han estado alguna pasada sin ser actualizadas.

Sin actualizar rango: Son todas las entradas que al final del rango aún no se han recogido.

Errores precisión: Número de errores de precisión en un rango. Se suman las celdas de Error precisión que contenían un SI. El máximo será de 8 por rango.

Errores cobertura pasada: Número de pasadas en las que ha habido error de cobertura. El máximo será de 8 por rango.

Errores cobertura rango: Si hay entradas sin actualizar en el rango valdrá 0. En caso contrario valdrá 1.

Recuperaciones: Número total de entradas actualizadas en el rango. Se suman todas las entradas de la fila Recupera.

Aciertos: Númerode veces que se lanza el Crawler y recupera al menos una entrada. El máximo será de 8 por rango.

Y los totales por día, que aparecen a la derecha de la página, son:

Pasadas: Número de pasadas que se han realizado a lo largo de un día.

Aciertos: Es el número de veces que se ha visitado la página y se han obtenido entradas. El máximo es 16 por día.

Errores precisión: Es la suma de todos los errores de precisión. Máximo 16 por día.

Recuperaciones: Son el número de entradas que se han podido recuperar en el día.

Errores cobertura pasada: Suma una unidad por cada periodo que hubiera errores de cobertura. El máximo también es 16 por día.

Errores cobertura rango: Si ha habido Sin actualizar rango vale 1, si no vale 0. Como máximo habría 4 por día.

Sin actualizar pasada: Número de entradas que se han quedado sin recuperar en alguna pasada durante el día.

Sin actualizar rango: Número total de entradas que se han quedado sin recuperar en algún rango durante el día.

Con la hoja de plantilla explicada, se podría ya crear una hoja sobre el registro siguiente. El perteneciente al día 3. En este libro se encuentran hechas hasta el día 31, pero se va a explicar el procedimiento con el que se hizo la hoja 3.

Nota: Hay que remarcar, que lo que en el libro Excel se ha denominado **Sin actualizar rango** es equivalente al **error de cobertura (e_c)** que se calcula como variable significativa, mientras que Errores de cobertura, ya sean de pasada o de rango dentro del libro, sólo esclarecen si hubo algún error de este tipo en alguna pasada o rango.

4.8 Creación de hojas a partir de la plantilla

Se crea una nueva hoja en el libro y se modifica el nombre a “Dia3”. Hay que tener en cuenta que las hojas de cálculo de visitas tienen que tener como nombre “Dia” más el número del día que se quiere calcular. Esto es debido a que en la hoja de resultados se usa “Dia” como parte del nombre para calcular un rango.

A continuación se selecciona toda la hoja “Dia2” y se copia. Se pega en la hoja “Dia3” y se modifica el valor de la celda E3, escribiendo en su interior el valor del día evaluado.

Se pulsa en la tecla F9 para recalcular los datos y ya estaría la hoja actualizada.

Quedaría un último punto por contar acerca del libro Excel. La hoja “Resultados”. En ella se encuentran los totales de todos los días, calculados como el sumatorio de todos los valores de cada día.

Además es donde se modifican los valores de los coeficientes y de la cota de probabilidad.

También se encuentra otra tabla con los datos relativos a la precisión, la cobertura y la media ponderada.

Una vez explicado el contenido de la hoja Excel y su funcionalidad, se pasará a ver las variables observadas para realizar el ajuste de coeficientes en la función principal.

4.9 Resultados de la simulación

Se han realizado 10 simulaciones con ocho valores diferentes para los coeficientes. Los resultados obtenidos se pueden observar en la siguiente tabla:

Tabla 1. Simulaciones en Excel.

prueba	alfa	beta	delta	pasadas	precisión	cobertura	media ponderada
1.- α=0,20; β=0,30; δ=0,50	0,20	0,30	0,50	60,70	0,7088	0,6481	0,6906
2.- α=0,15; β=0,25; δ=0,60	0,15	0,25	0,60	60,00	0,7184	0,6511	0,6982
3.- α=0,30; β=0,20; δ=0,50	0,30	0,20	0,50	**57,80**	0,7118	0,6198	0,6842
4.- α=0,50; β=0,30; δ=0,20	0,50	0,30	0,20	59,10	0,6893	0,6154	0,6671
5.- α=0,20; β=0,50; δ=0,30	0,20	0,50	0,30	63,70	0,6912	0,6609	0,6821
6.- α=0,10; β=0,25; δ=0,65	0,10	0,25	0,65	60,00	**0,7203**	0,6514	**0,6996**
7.- α=0,15; β=0,60; δ=0,25	0,15	0,60	0,25	64,20	0,6935	**0,6684**	0,6860
8.- α=0,15; β=0,40; δ=0,45	0,15	0,40	0,45	61,80	0,6987	0,6547	0,6855
Mejor valor				**57,80**	**0,7203**	**0,6684**	**0,6996**
Modelo anterior				**992,00**	**0,1435**	**1,0000**	**0,4004**

Además de los resultados obtenidos para el Crawler priorizado, puede observarse que en la parte inferior están los resultados para el Crawler según el modelo anterior.

En verde se han resaltado los mejores resultados para cada caso específico. Se corresponden con las medias de las 10 simulaciones realizadas, llamadas ensayos en la hoja Excel. Se han dejado los datos relativos a estos ensayos en hojas separadas para posibles pruebas. Para realizarlas no hay más que copiar el contenido de una de estas hojas y pegarlo en la hoja "Hoja1". Al pulsar F9 se recalcularán los resultados para las variables actuales, que se encuentran en la hoja "Resultados".

Para cada simulación se han ido modificando los valores de las variables y copiando los resultados en una tabla creada específicamente para ello en la hoja "Ajustes". Una vez rellena la tabla se ha copiado en su totalidad en el apartado correspondiente a su ensayo en la hoja "Totales". En esta hoja ha sido donde se ha calculado la media final.

Los valores a tener en cuenta para la medición han sido:

Pasadas → Número de pasadas totales del Crawler en el proceso total de los 31 días.

Precisión → Relación de aciertos y cantidad de pasadas para el proceso total de los 31 días.

Cobertura → Relación de entradas recuperadas y total de entradas disponibles para el proceso total de los 31 días.

Media ponderada → Relación de la precisión y la cobertura, con los valores ponderados al 70% y al 30% respectivamente, para el proceso total de los 31 días.

Se pueden observar los resultados de los diez ajustes realizados con las ocho variables desglosadas en el siguiente gráfico:

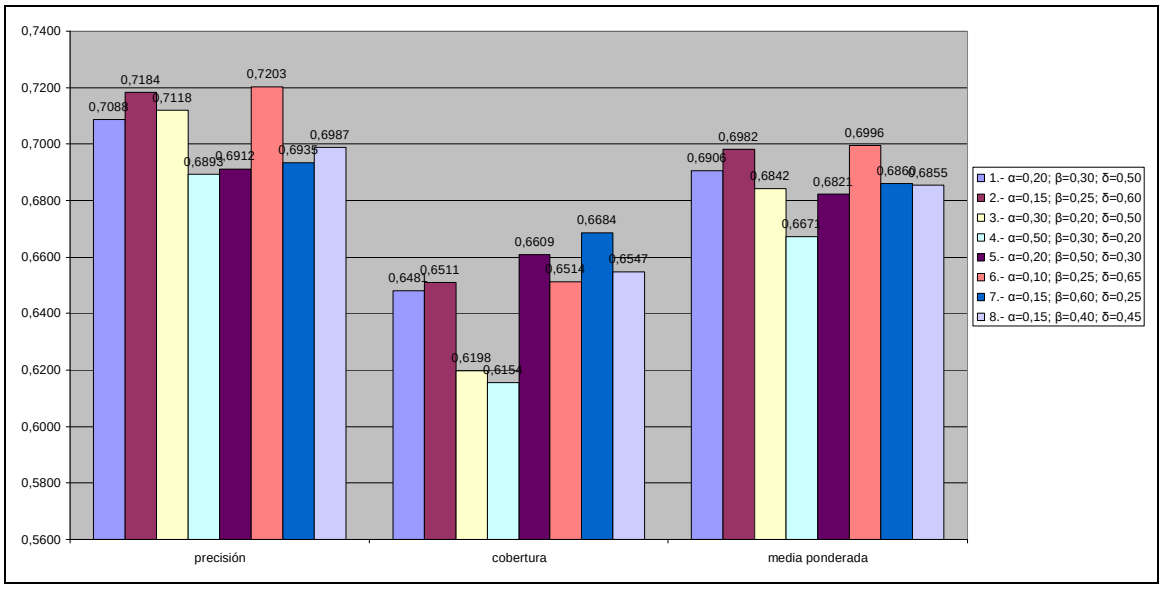

Figura 6. Figura con los resultados de los ajustes.

Puede observarse que para las variables precisión y media ponderada destaca la serie 6, con el color salmón, mientras que en la cobertura destaca la serie 7, de color azul. Más adelante se verá qué valores toman estas series para los coeficientes buscados y cuáles serán finalmente los escogidos.

Debido a que las pasadas generan números mucho mayores que los mostrados no se ha incluido en este gráfico. Pero a continuación se han desglosado cada una de las valoraciones en un gráfico individual para su mejor observación.

El número total de pasadas realizadas con cada ajuste ha sido de:

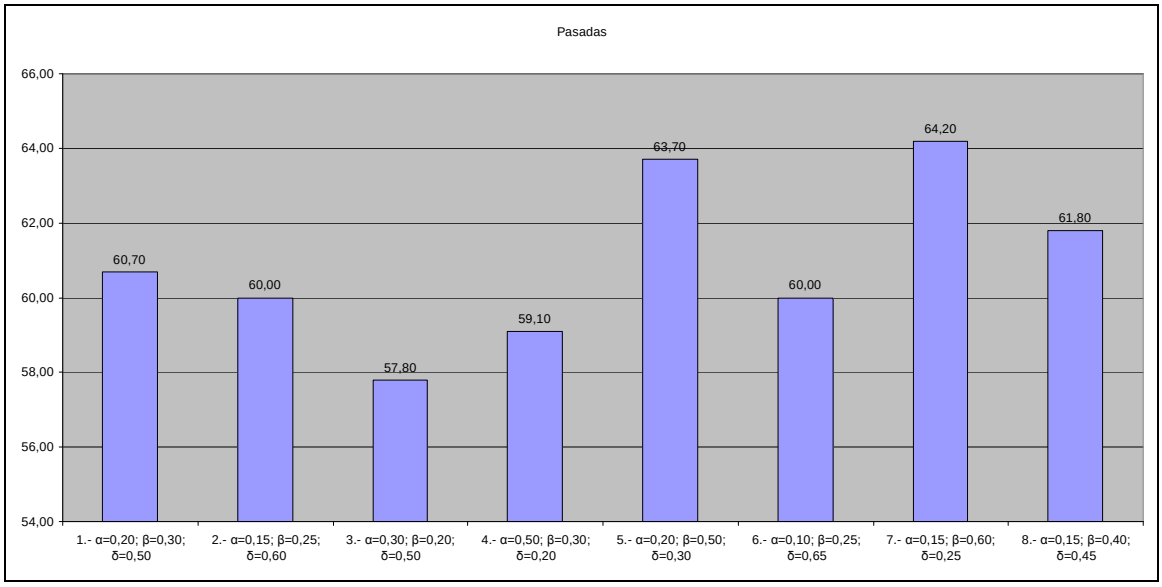

Figura 7. Figura con los resultados de las pasadas.

Este valor será mejor cuanto más pequeño. Se puede observar que el ajuste número 3 es el que menos pasadas realiza en total.

Los resultados para la precisión son:

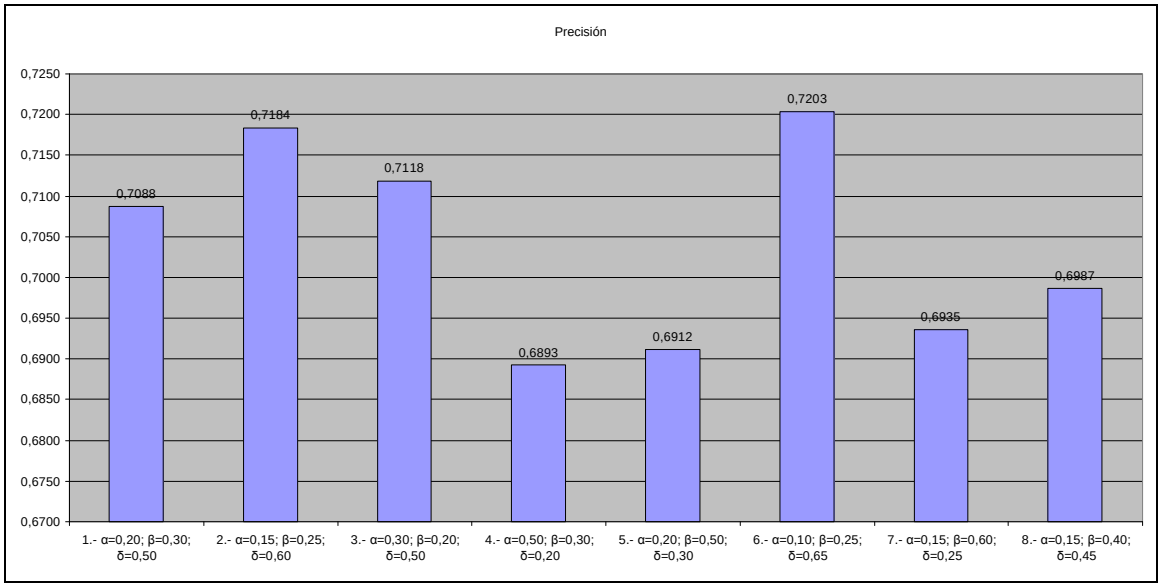

Figura 8. Figura con los resultados de precisión.

Este valor será mejor cuanto más grande. Se puede observar que el ajuste número 6 es ahora el que parece ser la mejor elección. Es el ajuste que más veces recupera entradas con respecto a sus visitas.

Los resultados para la cobertura son:

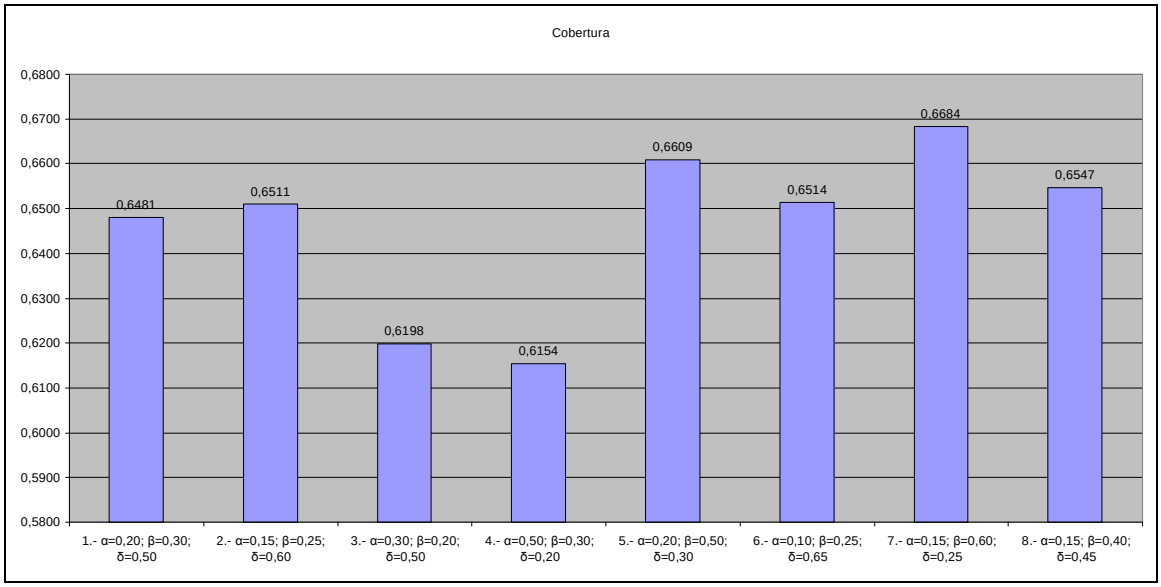

Figura 9. Figura con los resultados de los errores de cobertura.

Este valor será mejor cuanto más grande. En este caso es el ajuste número 7 el que dispone de un mejor resultado para la cobertura. Por lo tanto en este caso es dicho ajuste el que recupera más entradas en el periodo estudiado.

Los resultados para las medias ponderadas son:

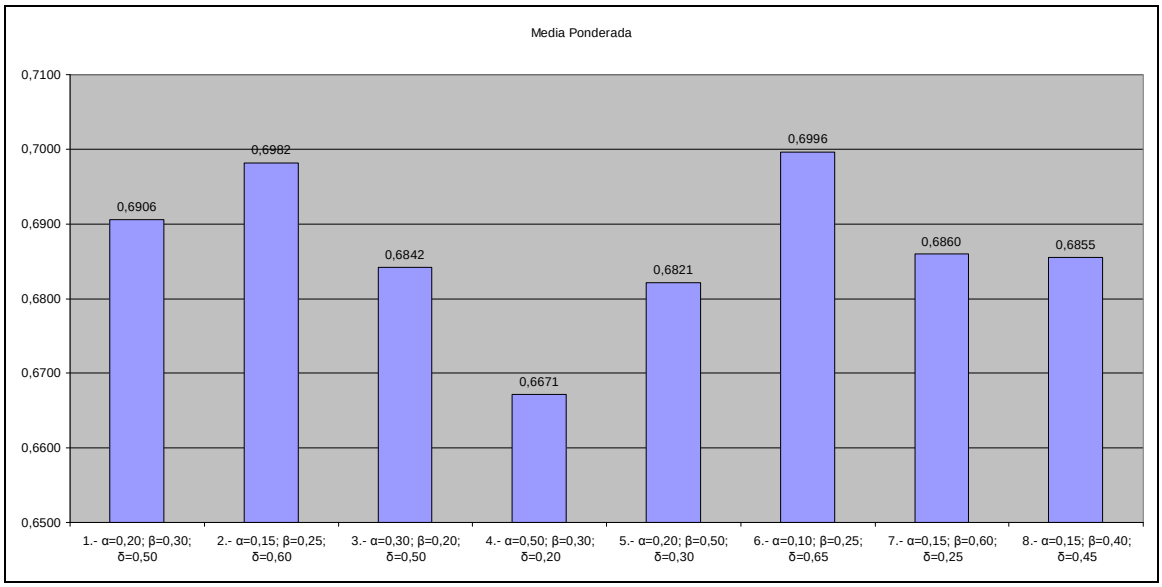

Figura 10. Figura con los resultados de las medias ponderadas.

Este valor será mejor cuanto más grande. En este caso también es el ajuste número 6 es el que mayor media ponderada posee.

Según los datos visualizados, los ajustes 3, 6 y 7 son los que mejores estimaciones aportan. Aunque también es cierto que hay un ajuste muy cercano al número 6 en todas las valoraciones, que sería el número 2. Por lo tanto habrá que elegir entre uno de ellos para configurar los coeficientes de la ecuación principal del modelo.

Tabla 2. Detalle de la simulación con los mejores valores.

prueba	alfa	beta	delta	pasadas	precisión	cobertura	media ponderada
2.- α=0,15; β=0,25; δ=0,60	0,15	0,25	0,60	60,00	0,7184	0,6511	0,6982
3.- α=0,30; β=0,20; δ=0,50	0,30	0,20	0,50	**57,80**	0,7118	0,6198	0,6842
6.- α=0,10; β=0,25; δ=0,65	0,10	0,25	0,65	60,00	**0,7203**	0,6514	**0,6996**
7.- α=0,15; β=0,60; δ=0,25	0,15	0,60	0,25	64,20	0,6935	**0,6684**	0,6860

Puede observarse qué hay datos a favor de uno y otro ajuste, pero no son demasiado grandes las diferencias. En principio se podría optar por la opción 6, puesto que su media ponderada y su precisión son las mejores. Son, además, las

dos variables que más interesan. Así que observando que la variable cobertura tampoco es demasiado bajas en comparación con la mejor de todos los ajustes, habría que escoger el número 6.

En cuanto al número de pasadas, el ajuste 6 es el segundo mejor. La diferencia es de 2,20 pasadas cada mes (según el ciclo de 31 días que se ha seguido). No parece muy significativo que se realicen 2 pasadas más en todos estos días. Si se tomara como base un año se podría ajustar el valor a $2{,}20 \times 12 = 26{,}4$ pasadas al año (considerando todos los meses de 31 días, claro).

Sobre la cobertura se puede observar que hay una diferencia de algo menos de 2 puntos porcentuales con respecto al ajuste 7. Esta variable se corresponde al porcentaje de entradas recuperadas del total de disponibles, por lo tanto el ajuste 7 recuperaría el 66,84% de las que genera un canal, mientras que el ajuste 6 recuperaría el 65,14%. Realmente no es exacto este indicador, puesto que se refiere a las recuperaciones en el rango y día determinado en que se genera la entrada. Es decir, la entrada se genera el día 1 pero el Crawler no pasa por el canal, por lo tanto no se recupera. Ahí se añadiría un error de cobertura. Sin embargo, esta entrada es más que probable que se recupere al día siguiente con otro lanzamiento del Crawler, o incluso el mismo día pero en otro rango. Por tanto, este indicador es menos preciso que el indicador de precisión. Es por esta razón por la que se ha ponderado a favor de la precisión en detrimento de la cobertura.

Teniendo todo esto en cuenta, es asumible escoger los valores para los coeficientes de la función principal que proporciona el ajuste 6, con lo que la función queda:

$$\lambda = 0{,}10w + 0{,}25m + 0{,}65h$$

Una vez se implemente la funcionalidad del modelo dentro de la aplicación java se podrán realizar simulaciones con más datos para contrastar estos valores y, si fuera necesario, modificar el ajuste.

Capítulo 5

Implementación

5.1 Introducción

El sistema actual realiza visitas a las páginas especificadas en un fichero OPML de entrada. Si recupera entradas en estas visitas, almacena la información de las mismas en ficheros RSS de salida. Una vez realizado el proceso y transcurrido el tiempo determinado entre pasadas, el Crawler vuelve a leer el mismo fichero de entrada y vuelve a realizar el mismo procedimiento.

El esquema general del proceso completo sería el siguiente:

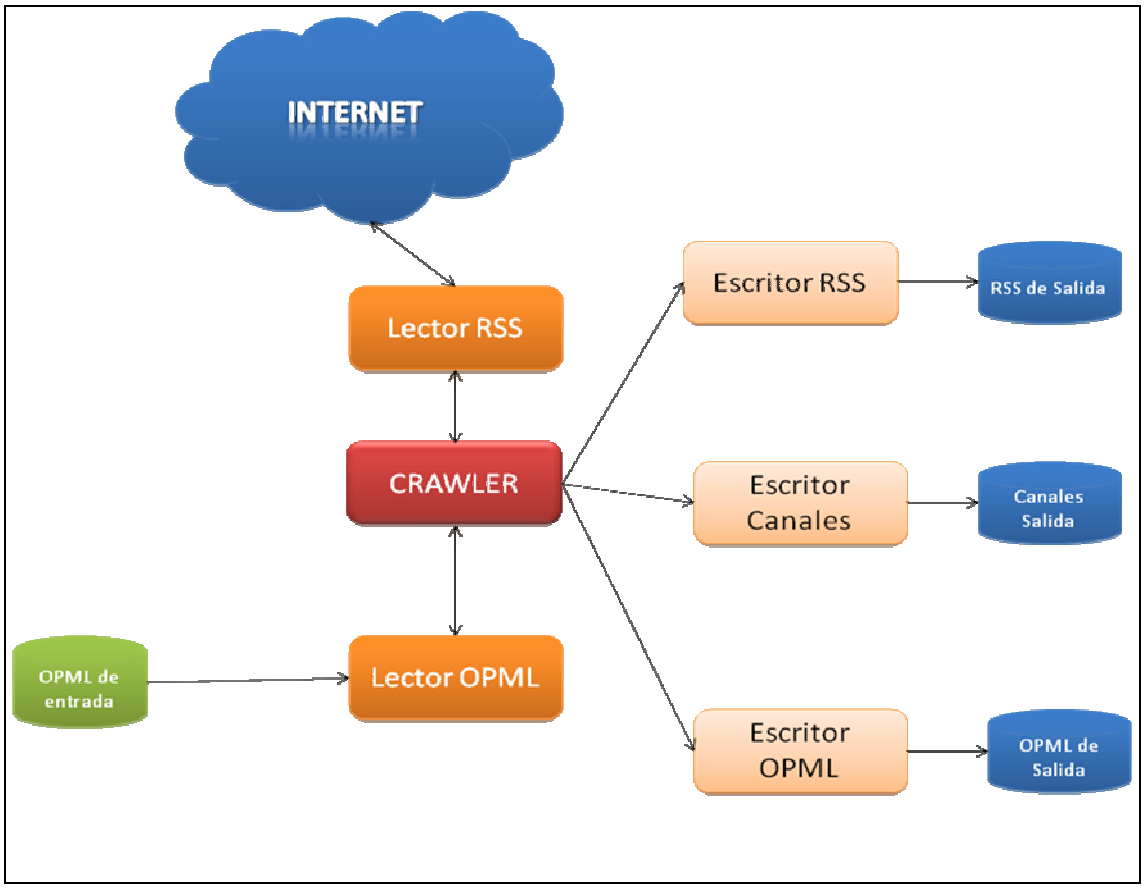

Figura 11. Diagrama del sistema actual.

5.2 Módulo de estadísticas

El modelo priorizado que se ha diseñado, se implementará como un módulo complementario al Crawler actual, y se llamará módulo de estadísticas. Este módulo es el encargado de analizar los datos almacenados en la base de datos y hallar los valores de λ para cada uno de los canales.

Para cada canal agrupa las entradas recuperadas relativas al mismo día de la semana que la fecha de lanzamiento. A continuación divide las entradas en los distintos rangos horarios según su fecha de edición. Por último calcula las entradas válidas según los límites definidos en la configuración, que se

corresponden con d, m y h. Esto resulta en el valor de λ para cada canal, que se almacenará, si supera la cota de probabilidad, en la tabla de visitas para que el Crawler visite la Web cuando sea necesario.

Las visitas se generan al comienzo del día y son válidas para este día en concreto. La tabla de visitas se puede actualizar a lo largo del día puesto que cuando se visita una página se pueden o no recuperar entradas y, por tanto, es posible que modifique el comportamiento del próximo lanzamiento del Crawler.

El esquema del proceso con el módulo de estadísticas quedaría:

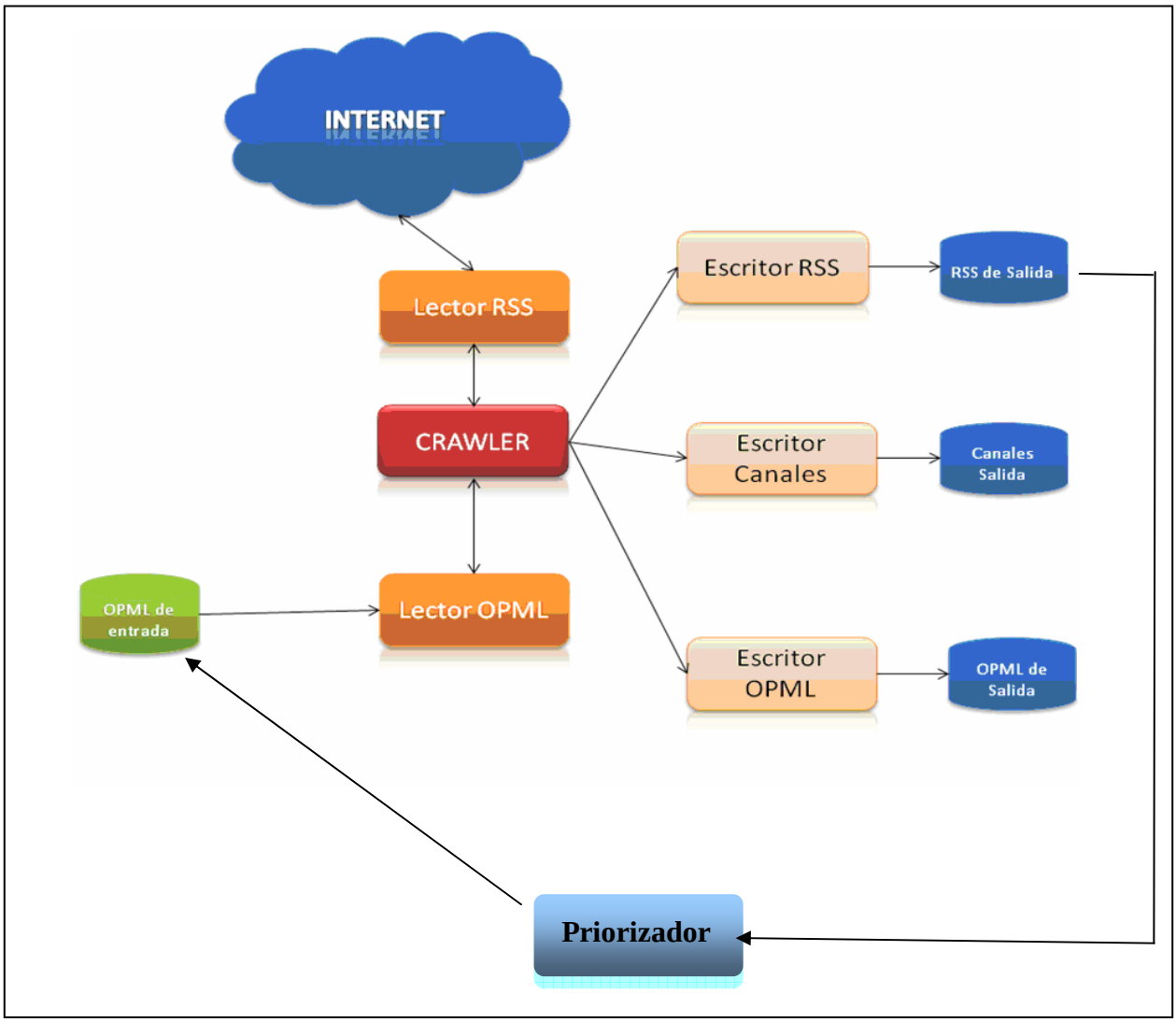

Figura 12. Diagrama del sistema priorizado.

Puede observarse cómo se ha introducido el módulo de estadísticas (con el nombre de Priorizador) entre la salida del escritor RSS, que generará un fichero

RSS, y el OPML de entrada. El módulo modificará el fichero OPML para que genere en cada pasada la lista de sitios que se visitarán.

5.3 Integración

El módulo de estadísticas estará incrustado entre la escritura del fichero de salida y la lectura de los ficheros de entrada del Crawler.

En el ciclo de escritura de las entradas recuperadas, que es cuando se están rellenando los ficheros RSS de salida, el módulo de estadísticas irá insertando las entradas recuperadas por el Crawler en la Base de Datos de la aplicación.

Una vez concluído un día completo, se lanzará un proceso automático que tratará los datos de las entradas recuperados en el propio día y los datos anteriormente existentes en la BD. Es aquí donde se calcularán los valores de la ecuación principal para las visitas del día siguiente.

Las visitas se almacenarán en la BD en una tabla creada para esta función. La creación del fichero de entrada OPML se realizará leyendo los datos almacenados en dicha tabla.

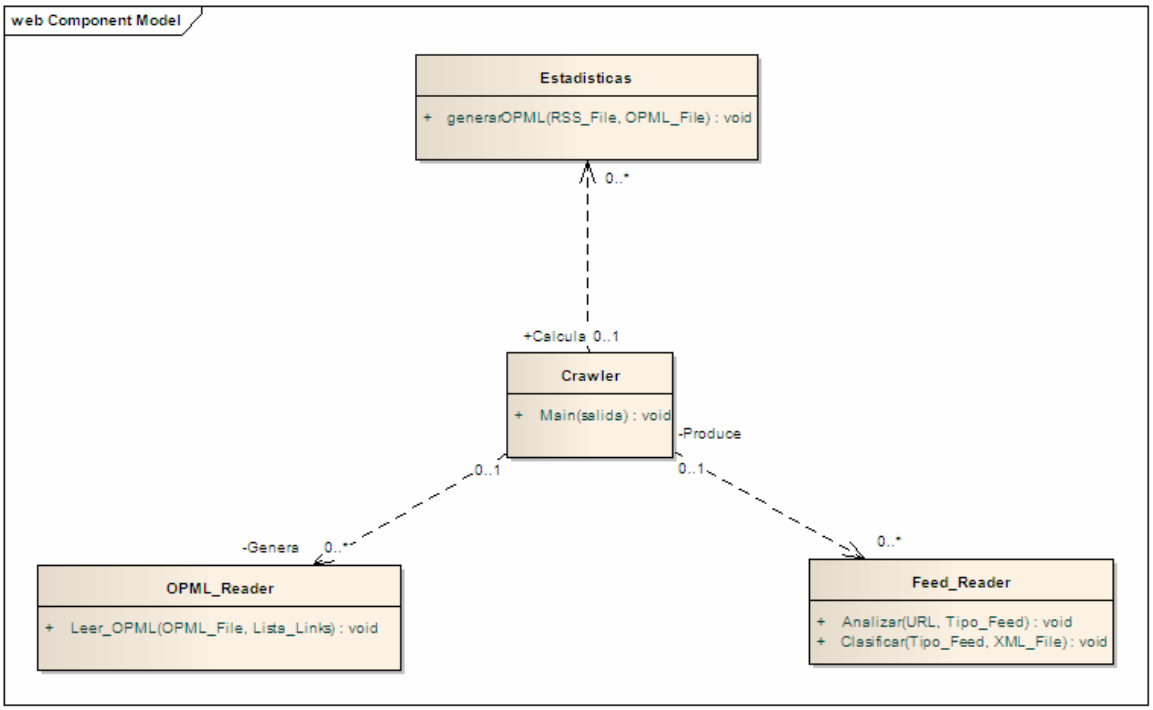

Figura 13. Diagrama de clases del sistema priorizado.

5.4 Pruebas

Se han realizado unas pruebas de simulación con los datos de los ensayos establecidos en el capítulo de simulación del modelo. Estos datos son para realizar la simulación en Excel pero se ha querido comprobar el funcionamiento del modelo implementado en java con los mismos datos que se había diseñado para comprobar que el comportamiento fuera el esperado.

Para ello se han utilizado sólo las pruebas con las variables elegidas únicamente, que se han evaluado con los diez ensayos realizados en el libro Excel.

Función principal: $\lambda = \alpha w + \beta m + \delta h$, en donde:

α= 0,10; β= 0,25; δ= 0,65; lim_min=1; lim_med=4; lim_max=24. El rango es de 2 horas y el tiempo entre pasadas son 15 minutos.

Los resultados obtenidos han sido:

Tabla 3. Resultados de los ensayos.

	Pasadas		Precisión		Cobertura		Media ponderada	
Simulación	**Excel**	**Java**	**Excel**	**Java**	**Excel**	**Java**	**Excel**	**Java**
Ensayo 1	53,00	53,00	0,6604	0,6604	0,6358	0,6358	0,6530	0,6530
Ensayo 2	54,00	55,00	0,7593	0,7455	0,6358	0,6398	0,7223	0,7137
Ensayo 3	59,00	60,00	0,7119	0,7167	0,6415	0,6478	0,6908	0,6960
Ensayo 4	80,00	81,00	0,7625	0,7531	0,7196	0,7196	0,7496	0,7430
Ensayo 5	47,00	47,00	0,7660	0,7660	0,5974	0,5974	0,7154	0,7154
Ensayo 6	67,00	67,00	0,6716	0,6716	0,6765	0,6725	0,6731	0,6719
Ensayo 7	53,00	53,00	0,7547	0,7547	0,6145	0,6258	0,7126	0,7160
Ensayo 8	77,00	77,00	0,6623	0,6623	0,6748	0,6748	0,6661	0,6661
Ensayo 9	51,00	51,00	0,7255	0,7255	0,6188	0,6266	0,6935	0,6958
Ensayo 10	59,00	59,00	0,7288	0,7288	0,6989	0,6989	0,7198	0,7198

5.4.1 Análisis de pruebas

Como se puede observar en las tablas de los ensayos, los resultados con la simulación en Excel y en Java son prácticamente idénticos. Hay ligeras variaciones que son debidas a la suposición de recogida de entradas que se hace en el propio Excel, cuya eficiencia en la programación no es la misma que la que se ha podido hacer con Java.

Vistos los resultados, se puede confirmar que son los esperados a partir de las simulaciones. El haber obtenido valores muy dispares en ambos entornos, habría ocasionado la variación de los coeficientes de la fórmula principal o de otros valores implicados. Como la resolución de las pruebas es favorable al modelo diseñado, se realizará la ejecución con datos reales con la implementación desarrollada.

Capítulo 6

Experimentación

6.1 Evaluación con datos reales

Para comprobar el funcionamiento del modelo priorizado aplicado en un entorno real, se han recuperado entradas de sitios Web reales durante un periodo de tiempo, a fin de conseguir datos suficientes para comprobar el funcionamiento de dicho modelo.

Los siguientes apartados explican la realización de las pruebas y analizan los resultados obtenidos.

6.2 Recuperación de entradas

Para la obtención de datos reales, se han incluido en el fichero de entrada OPML un total de 10 blogs y 10 periódicos de ámbito político. Se ha estado lanzando la aplicación con el modelo antiguo para recuperar una cantidad de datos significativa y así realizar las pruebas con unas estadísticas lo más completas

posible, puesto que el modelo priorizado se basa en los datos recuperados anteriormente.

El tiempo empleado en la recuperación de entradas de las Webs anteriormente indicadas ha sido de aproximadamente tres meses. Durante el proceso de recuperación de datos se han producido ciertos problemas con el ordenador que hacía las veces de lanzador del Crawler. Dicho ordenador ha estado expuesto a algunos virus y se han producido reinicios inesperados del sistema, perdiendo la posibilidad de lanzar el Crawler durante algunos periodos de tiempo, los cuales producian la inestabilidad de los datos por no recuperar los de algunos días. Por ello se ha considerado el utilizar el máximo intervalo de datos continuados que se ha conseguido en este tiempo de ejecuciones del Crawler. Se han conseguido obtener 39 días seguidos de actualizaciones. Es por ello que se han realizado las pruebas con datos de 30 días y la priorización del resto de días.

Es importante destacar que la ejecución de pruebas con datos reales no ha supuesto el lanzamiento del modelo priorizado durante el periodo establecido, puesto que para ello se necesitaría disponer de los datos continuados hasta el día actual y comenzar con dicho lanzamiento sin interrupciones, a ser posible. Por lo tanto se ha realizado la simulación en java de lo que sería la ejecución en real de los datos que se han recuperado en la totalidad del tiempo de ejecución.

Se han obtenido entonces las visitas que se tendrían que realizar y se han calculado como tal, comprobando si se recuperarían entradas o si habría errores de precisión cobertura.

Debido a que el modelo se basa en un sistema de datos de hasta un semestre completo, y los datos obtenidos apenas sí llegan al mes, se ha decidido realizar dos pruebas distintas. Una de ellas con los datos aportados en el diseño del modelo, en los que la variable lim_max se correspondería a 24 semanas de datos, pero en la que sólo habría datos durante 4. La siguiente prueba se ha realizado modificando los valores de los límites del modelo, teniendo lim_min=1, lim_med=2 y lim_max=4, para que dispongan de todos los datos que abarca cada variable.

Los datos que resulten de la segunda prueba podrían incurrir en un mayor error que la primera debido a que los coeficientes de la fórmula principal se han afinado según los valores originales de lim_min, lim_med y lim_max.

Los sitios visitados durante el periodo de ejecución han sido:

Tabla 4. Sitios visitados.

URL	Descripción
http://feeds.feedburner.com/escolar	Blog de Ignacio Escolar
http://cesar.lasideas.es/?feed=rss2	Blog de César Calderón
http://blogs.20minutos.es/arsenioescolar/feeds/rss2	Blog de Arsenio Escolar
http://www.marcvidal.cat/espanol/atom.xml	Blog de Marc Vidal
http://www.guerraeterna.com/index.xml	Blog de Íñigo Sáenz de Ugarte
http://feeds.feedburner.com/desdeelexilio/BPuO	Blog político (Desde el Exilio)
http://franciscopolo.com/feed/	Blog de Francisco Polo
http://feeds.feedburner.com/Manifestometro	Blog de recuento de asistentes en manifestaciones
http://www.javierortiz.net/jor/feed.xml	Blog de Javier Ortiz
http://www.moscasenlasopa.net/blog/?feed=rss2	Blog de Javier Mesonero
http://rss.elmundo.es/rss/descarga.htm?data2=8	Periódico digital El Mundo (España)
http://rss.feedsportal.com/c/805/f/535766/index.rss	Periódico digital Público (España)
http://www.ELPAIS.com/rss/feed.html?feedId=11	Periódico digital El País (Política)
http://rss.elmundo.es/rss/descarga.htm?data2=4	Periódico digital El Mundo (Portada)
http://www.20minutos.es/rss/	Periócido digital 20minutos
http://www.abc.es/rss/feeds/abc_Nacional.xml	Periódico digital ABC (Nacional)
http://www.larazon.es/rss	Periódico digital La Razón
http://www.adn.es/rss/politica/	Periódico digital ADN (Política)
http://www.abc.es/rss/feeds/abc_Opinion.xml	Periódico digital ABC (Opinión)
http://www.elconfidencial.com/rss/opinion.xml	Periódico digital El Confidencial

6.3 Prueba con resultados reales

6.3.1 Modelo diseñado

La primera prueba realizada se ha efectuado sobre el modelo diseñado con los valores calculados en la experimentación. Los datos de configuración del Crawler son:

Tabla 5. Valores de la primera prueba.

Variable	Valor
Lim_min	1
Lim_med	4
Lim_max	24
α	0,10
β	0,25
δ	0,65

Los resultados obtenidos han sido:

- Se han actualizado 1644 registros.
- Nº pasadas= 387, que es el número de veces que se visita un canal.
- Nº aciertos= 245, que es el número de veces que se visita un canal y se recupera al menos una entrada.
- Nº errores= 142, que es el número de veces que se visita un canal y no se recupera ninguna entrada.
- Nº entradas perdidas= 826, que es el número de entradas que no se han recuperado en su rango específico porque no se lanzó el Crawler.
- Entradas sin recuperar= 123, que es el número de entradas que no se han recuperado al finalizar todos los lanzamientos del Crawler (Se recuperarían en los sucesivos lanzamientos).

La precisión es de $\frac{aciertos}{pasadas} = \frac{245}{387} = 0{,}633$, mientras que la cobertura es $\frac{actualizadas}{actualizadas + perdidas} = \frac{1644}{1644 + 826} = 0{,}6655$. La media ponderada es de $0{,}7 * precisión + 0{,}3 * cobertura = 0{,}7 * 0{,}633 + 0{,}3 * 0{,}6655 = 0{,}6427$.

A pesar de que el número de entradas que se han perdido es de 826, tan sólo se han quedado sin recuperar 123. Esto es debido a que las entradas perdidas lo son en un único rango, pero si el canal vuelve a visitarse en otro rango logra recuperar estas entradas. Las entradas sin recuperar, sin embargo, pertenecen a canales que no se han vuelto a visitar y que es posible que no se visiten. Esta deficiencia se corrige, como se ha explicado en capítulos anteriores, visitando las Webs que tienen una probabilidad por debajo de la cota mínima en horarios poco ociosos, para que por lo menos se recuperen y vayan aumentando la probabilidad de visita en sus rangos más activos.

Sin tener en cuenta los datos sobre precisión, el número total de entradas recuperadas es de $recuperadas = actualizadas + perdidas = 1644 + 826 = 2470$ y el porcentaje total de entradas recuperadas es de $\frac{recuperadas}{recuperadas + \text{sin } recuperar} = \frac{2470}{2470 + 123} = 0{,}9525$.

6.3.2 Modelo con Lim_max=4 semanas

Como ya se ha explicado anteriormente, para compensar la falta de datos se ha decidido realizar una segunda prueba con los valores de los límites modificados con respecto al diseño implementado. Los datos de la prueba son, por tanto:

Tabla 6. Valores de la segunda prueba.

Variable	Valor
Lim_min	1
Lim_med	2
Lim_max	4
α	0,10
β	0,25
δ	0,65

Los resultados obtenidos han sido:

- Se han actualizado 2050 registros.
- Nº pasadas= 1034, que es el número de veces que se visita un canal.
- Nº aciertos= 633, que es el número de veces que se visita un canal.
- Nº errores= 401, que es el número de veces que se visita un canal y no se recupera ninguna entrada.
- Nº entradas perdidas= 475, que es el número de entradas que no se han recuperado en su rango específico porque no se lanzó el Crawler.
- Entradas sin recuperar= 68, que es el número de entradas que no se han recuperado al finalizar todos los lanzamientos del Crawler (Se recuperarían en los sucesivos lanzamientos).

La precisión es de $\frac{aciertos}{pasadas} = \frac{633}{1034} = 0{,}6121$, mientras que la cobertura es $\frac{actualizadas}{actualizadas + perdidas} = \frac{2050}{2050 + 475} = 0{,}8118$. La media ponderada es de $0{,}7 * precisión + 0{,}3 * cobertura = 0{,}7 * 0{,}6121 + 0{,}3 * 0{,}8118 = 0{,}672$.

Se puede observar que con estos valores se han realizado más del doble de pasadas que en la primera prueba. Esto es debido a que la media de entradas para el límite máximo es más grande al disminuir su divisor (de 24 pasa a 4). Este aumento de las visitas podría ser el original y se podría comprobar si realmente se tuvieran los datos relativos a 6 meses de ejecución del Crawler.

Sin tener en cuenta los datos sobre precisión, el número total de entradas recuperadas es de $recuperadas = actualizadas + perdidas = 2050 + 457 = 2525$ y el porcentaje total de entradas recuperadas es de $\frac{recuperadas}{recuperadas + \text{sin } recuperar} = \frac{2525}{2525 + 68} = 0{,}9737$.

6.4 Análisis de resultados

Una vez realizadas las pruebas con la diferencia de valores entre ambas, se puede comparar el resultado obtenido con el calculado para la simulación en Excel.

Precisión
Mientras que para la simulación en Excel se obtuvo un resultado de 0,7203, en la primera prueba se ha obtenido 0,633 y en la segunda 0,6121.

Este valor de la precisión es algo menor del esperado. Posiblemente sea por la falta de datos suficientes para la realización de la ejecución con datos reales. Aún así, una de las posibilidades que ofrece el modelo es la de modificar los valores de los coeficientes, de manera que a lo largo del tiempo se afinen más y se consiga una mayor precisión.

Cobertura
El resultado de la cobertura es, sim embargo, mayor del esperado. Para la simulación se obtuvo un valor de 0,6514, mientras que para la primera prueba se ha obtenido 0,6655 y para la segunda prueba se ha aumentado hasta 0,8118. El aumento de la cobertura para la primera prueba es prácticamente insignificante. Para la segunda prueba puede ser debido al cambio del valor que se ha explicado anteriormente.

Media Ponderada
El resultado para la media ponderada también se ha reducido del 0,6996 de la simulación en Excel al 0,6427 de la primera prueba y el 0,672 de la segunda. Este valor se ha visto reducido precisamente por la disminución de la precisón. Es posible que si se tuvieran los datos relativos a 6 meses de ejecución, el valor de la media ponderada se aproximara más al obtenido en la simulación en Excel.

Faltaría determinar como hubiera sido la ejecución del Crawler sin priorizar en el transcurso de estos 9 días completos. Para calcular el número de accesos basta calcular que si el Crawler pasa 4 veces en una hora (cada 15 minutos) y se lanza las 24 horas durante los 9 días, visitando 20 Webs (entre periódicos y blogs):

$$Accesos = 4 * 24 * 9 * 20 = 17280 .$$

El número de aciertos se puede realizar con una consulta a la base de datos de las entradas recuperadas. El total de aciertos resulta 253. Por lo tanto, la precisión de $\frac{aciertos}{pasadas} = \frac{253}{17280} = 0{,}0146$. Y la cobertura es 1 siempre.

Es decir, que la comparación entre realizar la priorización de visitas y no realizarla implica que se puede pasar de un 1% de precisión en las visitas a un 63% sin que el impacto por pérdida de alguna entrada sea grave.

La media ponderada para el modelo anterior quedaría como $0{,}7 * precisión + 0{,}3 * cobertura = 0{,}7 * 0{,}0146 + 0{,}3 * 1 = 0{,}3102$ que es prácticamente la mitad que con el modelo priorizado.

Capítulo 7

Conclusiones

7.1 Resumen del modelo

El modelo desarrollado en el presente documento, se ha implementado como mejora del actual aplicativo denominado Crawler. En este modelo priorizado se han utilizado las entradas recuperadas con anterioridad por el Crawler como método de retroalimentación. De manera que se hace un análisis de en qué horarios es más probable que una página Web esté actualizada, y sea en ese horario en el que el Crawler visite dicha página.

Se ha dividido la estimación de pasadas en tres periodos diferentes (una semana, un mes y un semestre) para agrupar posibles datos homogéneos teniendo en cuenta los factores más comunes asociados a la vida de una página Web. Con estos tres valores se ha realizado una combinación lineal para disponer de un único valor de estimación, que será el que se siga para evaluar si una página debe ser visitada o no.

La fórmula principal del modelo es $\lambda = \alpha w + \beta m + \delta h$.

Cada día se recalculan las visitas que se harán a los canales almacenados en la base de datos.

El Crawler se seguirá lanzando cada 15 minutos (siempre que haya algún canal en el fichero de entrada OPML) pero sólo con los canales que haya especificado el módulo de estadísticas.

Los canales que no se vayan a visitar según el modelo priorizado, se almacenarán en una tabla de no visitados. Los canales que estén incluídos en esta tabla serán visitados en horarios poco ociosos para que no se pierda su actividad.

7.2 Comparación con el Crawler actual

La ventaja principal que posée el modelo priorizado respecto al Crawler actual es el rendimiento.

Mientras que con el Crawler actual, cada 15 minutos se visitan todas las páginas que haya almacenadas en el fichero OPML de entrada (y por consiguiente en la base de datos), con el modelo priorizado se visita una parte de los canales almacenados y en horarios determinados, no siempre.

La aplicación no dispone de unos sistemas capaces de asumir el lanzamiento del Crawler cada 15 minutos, visitando todas las páginas que se tienen almacenadas en la base de datos. Teniendo en cuenta que los canales van aumentando según se recuperan URLs de otros sitios en las propias entradas recuperadas. Debido a esto, es preferible que se puedan eliminar las pasadas innecesarias, que se sabe que no van a recuperarse resultados o en ocasiones puntuales.

Se podrá hacer crawling de muchas más fuentes debido a que ahora hay más precisión y, por lo tanto, menos trabajo del procesador. Con ello es posible mejorar el dato que aporta GoogleReader de 2.000 canales, puesto que en el modelo diseñado no se realizará la visita a todos los canales como norma general.

Al no tener que realizar los millares de visitas que se debieran en cada pasada, el sistema dispondrá de más tiempo de procesamiento para tratar la información recuperada. Y también tendrá tiempo el procesador para ser utilizado en otras tareas distintas.

La diferencia del uso del procesador puede comprobarse con la comparación de las ejecuciones de los datos reales con el modelo priorizado y las que se tendrían de haber lanzado el Crawler actual.

El número de accesos que se hubiera realizado en las pruebas con el Crawler actual sería de 17280. Mientras que con el modelo priorizado han sido 387 (con el límite máximo en 4 semanas han sido 1034 accesos). Esto da una relación de $\frac{accesosActual}{accesos\,\mathrm{Pr}\,iorizado} = \frac{387}{17280} = 0{,}0223$ que resulta en un 2,23%. O lo que es lo mismo, de cada 100 accesos que se realizan con el Crawler actual, apenas se realizan 2 con el modelo priorizado.

Si bien es cierto que se quedan sin recuperar 123 entradas, estas se recuperarían mediante el lanzamiento del Crawler en los horarios poco ociosos ya que se deben a sitios Web que no superan la cota de probabilidad, y serían incluídos en la tabla de no visitados.

7.3 Líneas futuras

El modelo priorizado que se ha expuesto soluciona el problema fundamental del Crawler actual. Sin embargo, se podría modificar el comportamiento del modelo para abarcar casos más específicos de ciertos sitios Web. Casos que serían incluso de carácter subjetivo. Como por ejemplo la importancia de ciertas páginas con respecto a otras.

Quizás al usuario final le interese especialmente una página Web y quiera asegurarse de no perder ninguna actualización o de tenerla lo más rápidamente posible.

O también podría ser posible que esté interesado en actualizaciones que sólo se hacen los sábados en un determinado sitio.

Se podría implementar para estos casos algún tipo de planificador en el que se pueda configurar que un determinado canal se visite con cada lanzamiento del Crawler. O todos los sábados. O que se visite al menos una vez en cada tango. Que entre un determinado horario se visite en todas las pasadas, etc.

Se podría implementar un proceso de modo que una vez generadas las estadísticas diarias, se recuperara de la base de datos la configuración de los canales que se hubiera definido, para que así que insertaran visitas en ese día según las características requeridas.

Con este proceso se aumentaría la funcionalidad propia del modelo priorizado, añadiendo un nuevo comportamiento complementario. Obviamente, las visitas que se crearan desde este nuevo proceso se añadirían a las de el módulo de estadísticas y, en caso de que los canales estuvieran en la tabla de no visitados, se eliminarían de esta tabla debido a su inclusión en las visitas con este nuevo proceso.

Capítulo 8

Costes y Presupuesto

8.1 Costes

En este capítulo se listan los costes asociados a los medios materiales y a los recursos humanos utilizados en el desarrollo de este proyecto.

Se ha planificado un calendario laboral que implica una jornada de cinco horas diarias. Aunque la línea de realización del proyecto ha sido irregular a lo largo de todo el proceso, se ha supuesto una dedicación continua. Por lo que la duración del proyecto y el número de jornadas refleja la medida real de las horas invertidas en el desarrollo del proyecto al completo.

La distribución de los trabajos realizados se ha divido en las siguientes fases.

- Fase de análisis: esta fase incluye además del análisis del modelo, toda la investigación previa realizada sobre el sistema anterior llamado Crawler, así como la generación de la documentación tanto de la memoria como del aplicativo. El computo de 35 días * 5 horas/día hacen un total de 175 horas.

- Fase de diseño técnico: esta fase incluye la parte concerniente al diseño del aplicativ. El computo de 15 días * 5 horas/día hacen un total de 75 horas.

- Fase de implementación y pruebas: esta fase incluye la implementación del sistema, la implementación del simulador en Excel, la definición de la batería de pruebas a realizar, la instalación del aplicativo y la realización de las pruebas. El computo de 30 días * 5 horas/día hacen un total de 150 horas.

El desglose presupuestario de costes de materiales y recursos materiales utilizados se presenta a continuación en la siguiente hoja de cálculo. Teniendo en cuenta la planificación realizada en jornadas de cinco horas diarias, el valor de un 'hombre/mes' se establece en cien horas de trabajo, habiéndose hecho los cálculos en base a este dato.

En los costes de equipos se han incluido solo los referentes al material utilizado para realizar el proyecto. El servidor en el que se instalará el aplicativo no se ha incluido, dado que lo provee la Universidad.

Según las cifras establecidas en la hoja de cálculo del desglose presupuestario del proyecto, se puede determinar que el presupuesto total de este proyecto asciende a la cantidad de trece mil novecientos cincuenta (13.950,00 €) euros.

Leganés a 01 de octubre de 2011

El ingeniero proyectista

Fdo. José Vicente Sevillano Martín

8.2 Presupuesto

La planificación y los costes descritos anteriormente pueden verse en una hoja con el presupuesto desglosado a continuación.

UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:

José Vicente Sevillano Martín

2.- Departamento:

Informática

3.- Descripción del Proyecto:

- Titulo: Modelo para la actualización eficiente de contenidos en un Crawler de ficheros RSS
- Duración (meses): 4
- Tasa de costes Indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):

13.950,00 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a titulo informativo)	Categoría	Dedicación (hombres mes) [a]	Coste hombre mes	Coste (Euro)	Firma de conformidad
		Analista Programador	3,5	2.694,39	9.430,37	
		Analista	0,5	4.289,54	2.144,77	
					0,00	
					0,00	
					0,00	
		Hombres mes	**4**	**Total**	11.575,14	

[a] 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable [d]
Ordenador portátil	750,00	100	3	60	37,50
Ordenador sobremesa	250,00	100	3	60	12,50
		100		60	0,00
		100		60	0,00
		100		60	0,00
					0,00
				Total	50,00

[d] Fórmula de cálculo de la Amortización:

$$\frac{A}{B} x C x D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
	Total	0,00

OTROS COSTES DIRECTOS DEL PROYECTO[e]

Descripción	Empresa	Costes imputable
	Total	0,00

[e] Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	**Presupuesto Costes Totales**
Personal	11.575
Amortización	50
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes Indirectos	2.325
Total	13.950

Capítulo 9

Glosario

- **API**: interfaz de programación de aplicaciones.
- **Canal**: Sitio identificado unívocamente que dispone de una determinada información asociada a él. En el caso de la sindicación, un canal puede ser una web. También denominado Feed.
- **Crawler**: Sistema de recuperación masiva de información.
- **Eclipse**: Entorno gráfico de desarrollo.
- **Entrada**: Cada una de las unidades de información de componen un canal.
- **Java**: lenguaje de programación orientado a objetos.
- **OPML**: lenguaje de marcado para el procesamiento de esquemas. Utilizado para agrupar y categorizar esquemas de recursos. Por ejemplo, URL.
- **RSS**: Formato de sindicación.

- **URL**: caso particular de URI, en este caso, la dirección de una web o un recurso en internet.

- **Web**: conjunto de documentos de hipertexto.

Capítulo 10

Bibliografía

9.1 Libros

En este apartado se exponen los libros que han sido las principales fuentes de información.

Daniel Peña, "Fundamentos de Estadística"
Alianza Editorial, 2001.

Bruce Eckel, *"Piensa en java"*
Madrid, 2007, McGraw-Hill

9.2 Enlaces

En este apartado se exponen las páginas Web que han sido las principales fuentes de información.

OPML: Comunidad sobre el estándar.
http://www.opml.org/

Lector RSS: Web informativa acerca de los lectores RSS.

http://www.rss.nom.es/lector-rss/

Feedreader: Web propia del proyecto.

http://www.feedreader.com/

GoogleReader: Web propia de la aplicación.

http://www.google.es/reader/

RSSOwl: Web propia del proyecto.

http://www.rssowl.org/

RSS BOARD: Web informativa sobre sindicación

Especificación: http://www.rssboard.org/rss-specification

RSS 0.92: http://www.rssboard.org/rss-0-9-2

RSS 0.91: http://www.rssboard.org/rss-0-9-1

Web.Resource.org: Especificación RSS 1.0

RSS 1.0: http://web.resource.org/rss/1.0/spec

Bing Liu: Libro en edición digital (Web Data Mining)

http://www.cs.uic.edu/~liub/WebMiningBook.html

Nutch: Motor de búsqueda open source

http://nutch.apache.org/